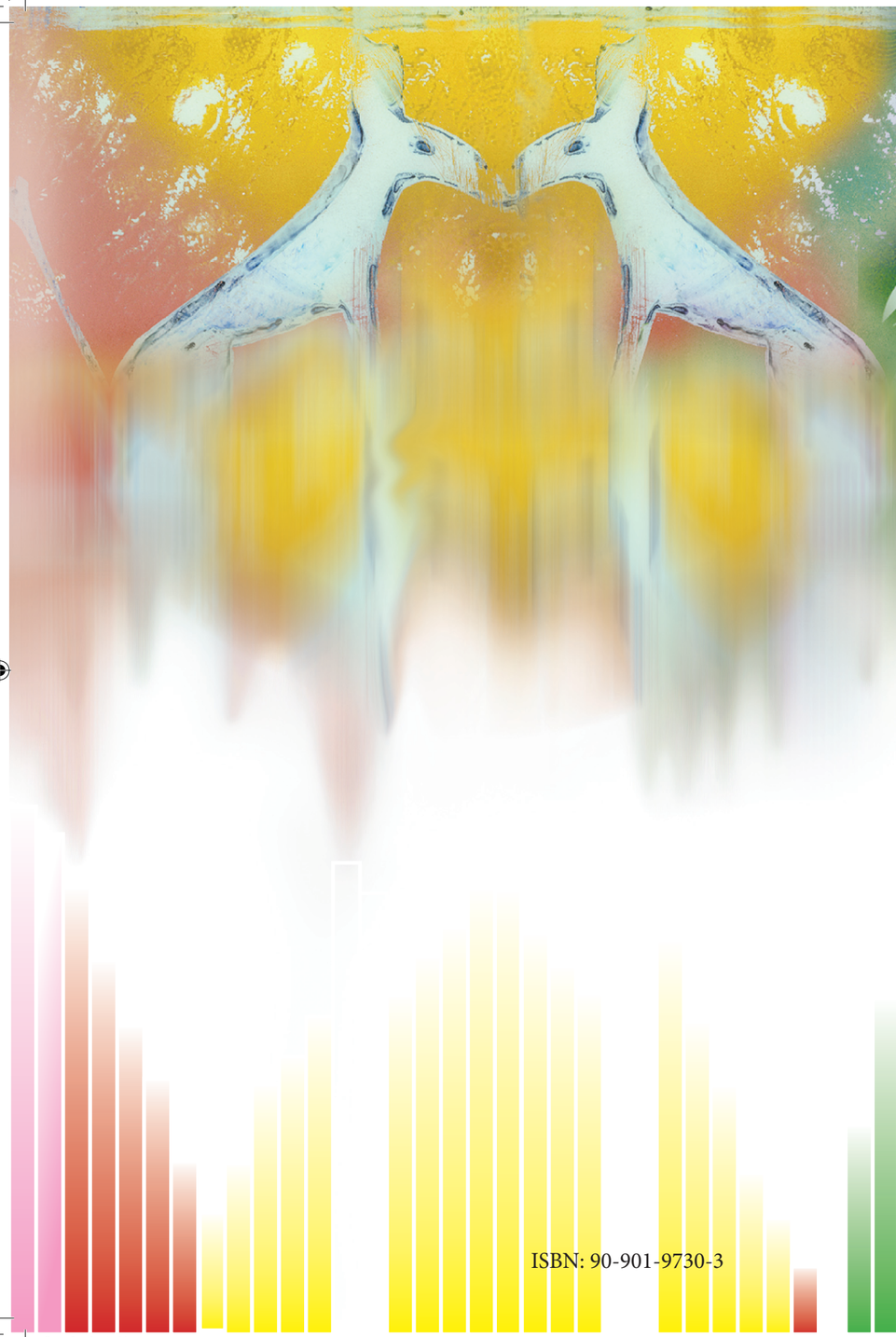


Human-Centered Content-Based Image Retrieval

Egon L. van den Broek

ISBN: 90-901-9730-3

Human-Centered Content-Based Image Retrieval

Egon L. van den Broek

Uitnodiging

voor de verdediging van mijn proefschrift op
woensdag 21 september 2005
om 13.30 uur in de aula van de Radboud Universiteit Nijmegen,
Comeniuslaan 2 te Nijmegen.
Borrel na afloop.

Feest op
zaterdag 24 september 2005
vanaf 21.00 uur in Diogenes,
Van Schaeck Mathonsingel 10 te Nijmegen.

Egon van den Broek
Wachterslaan 173
6523 RV Nijmegen
T: 06-48 69 67 18
E: egon@few.vu.nl

(kado? gewoon komen :)

Stellingen

behorende bij het proefschrift

Human-Centered Content-Based Image Retrieval

1. Hoe toegepast mag wetenschap worden als er experimentele data blijven bestaan die de toets der praktijk niet kunnen doorstaan? *(dit proefschrift)*
2. De techniek maakt sprongen die de wetenschap niet kan volgen. *(dit proefschrift)*
3. Zonder methoden geen onderzoek en zonder validatie hiervan geen wetenschap.
4. Eén van de grootste industriële problemen van deze eeuw wordt het gebrekkige informatiemanagment.
5. Een wiskundige kent perfectie, de andere stervelingen moeten het doen met "het best mogelijke".
6. Ik bedrijf wetenschap als ik inspiratie heb, en ik zorg er voor dat ik iedere ochtend om acht uur inspiratie heb. (geïnspireerd door Peter de Vries)
7. Het is een goede gewoonte om je onderzoek op te schrijven. Dat bespaart je de moeite om er iemand anders mee lastig te vallen. (geïnspireerd door Isabel Colegate)
8. Onderwijs is het ondergeschoven kindje van de Nederlandse universiteiten.
9. *Verleg je grenzen* moet worden geïnterpreteerd als: *Leer je grenzen kennen, accepteer ze en verleg ze indien mogelijk.*
10. There is only one way ...

Live today,
Plan for tomorrow,
Party tonight!

Egon L. van den Broek
Nijmegen, 3 augustus 2005

Human-Centered
Content-Based Image Retrieval

Egon L. van den Broek

This book was typeset by the author using LaTeX 2_ε.

Cover: Design and graphics by Wilson Design, Uden.

Printing: PrintPartners Ipskamp, Enschede - The Netherlands.

ISBN-number: 90-901-9730-3

Human-Centered Content-Based Image Retrieval

een wetenschappelijke proeve op het gebied van
de Sociale Wetenschappen

Proefschrift

ter verkrijging van de graad doctor
aan de Radboud Universiteit Nijmegen
op gezag van de Rector Magnificus prof. dr. C.W.P.M. Blom,
volgens besluit van het College van Decanen
in het openbaar te verdedigen op woensdag 21 september 2005
des namiddags om 1.30 uur precies

door

Egidius Leon van den Broek

geboren op 22 augustus 1974
te Nijmegen

Promotor:	Prof. dr. Ch.M.M. de Weert	
Co-promotoren:	dr. L.G. Vuurpijl dr. Th.E. Schouten	
Manuscriptcommissie:	Prof. dr. C.M. Jonker	
	Prof. dr. E.O. Postma	(Universiteit Maastricht)
	dr. M. Worring	(Universiteit van Amsterdam)
Paranimfen	*ing. ing. Peter M.F. Kisters, M.Sc.* *Eva M. van Rikxoort, M.Sc.*	

This research was supported by the Netherlands Organisation for Scientific Research (NWO) under project number 634.000.001.

The work described in this thesis has been carried out within the Cognitive Artificial Intelligence division of the graduate school NICI, Radboud University Nijmegen.

Contents

1

General introduction

This thesis is the result of work done within the NWO ToKeN project: Intelligent Content-Based Image Retrieval (CBIR). The project's name is Eidetic. According to Webster's dictionary [181], Eidetic means: "marked by or involving extraordinarily accurate and vivid recall especially of visual images". The latter should be achieved by intelligent CBIR techniques. Intelligence is the ability to learn, understand, or deal with new or trying situations [181] (e.g., recognizing objects in images). CBIR techniques aim to describe the content of the image material. We will approach the latter from both the perspective of human vision and of computer vision.

1.1 Human vision versus Computer vision

Par excellence, humans can function well in complex environments, which provide them with a range of multi-modal cues. So far, entities that comprise artificial intelligence cannot function properly if at all in such an environment. Therefore, human intelligence and, moreover, human cognitive capabilities are the baseline for the development of intelligent software applications, such as CBIR engines.

We know that the human visual system is powerful. "It endows us with the ability to recover enormous details about our visual environment. Nevertheless, as a plethora of visual illusions demonstrates, it is far from accurate. Today we interpret many of these illusions not as an expression of the limits or failures of the visual system. Rather, they are the results of a highly developed and optimized representational process in which the visual system does not simply provide an internal one-to-one copy of the external visual world. Instead, the visual system is equipped with specific encoding mechanisms to optimize the use of precious processing resources by enhancing relevant features and providing only a sketchy representation of the less relevant aspects of our visual environment." [297] In an ideal situation, computer vision should have similar characteristics.

From literature in neuroscience and psychology, it is known that the human visual system utilizes features such as color, texture, and shape in recognizing objects, the environment, and photos or paintings [93, 180, 239]. In addition, phenomena such as occlusion and subjective contours are topic of research. Research on these features and phenomena is merely done in controlled experimental environments. Control up to a large extent has the advantage that the influence of one factor on the percept can be examined. The drawback is that the ecological validity of such research is often questionable.

Especially on the long run, fundamental research is of great importance for application centered research, despite all its limitations. With such research, principles of human information processing can be unraveled step by step. Where in human vision research, this approach is the standard, for computer vision pragmatic considerations dominate. The

latter is not surprising. The core aim of computer vision in general is not to model human vision but to be inspired by human vision: An efficient approach for practical problems within a limited domain.

Let us provide two examples which illustrate the strength of human vision: (i) a painting cannot be described with computer vision techniques like humans can, who can abstract from detail and 'feel' the expressed emotion through their senses in combination with associations derived from memory [315] and (ii) although computer vision techniques are utilized for the automatic classification of mammography images, the performance of the classifiers developed so far is far from good. Consequently, most medical imaging techniques are used as computer aided diagnosis instead of replacing the human diagnosis [278].

On the other hand, for certain tasks computer vision algorithms outperform human vision by far. When the exact percentage of a specific type of red (a R, G, B value) in an image has to be determined, a computer can provide an exact percentage, where a human can only estimate it. The latter is more complicated when a fuzzy description of a color (e.g., red) is provided and when the results are judged by humans. Then the question arises: what is red? Are certain pixels in all circumstances judged as red by humans or can they be judged as being black, for example, when the environment or the light source changes? The latter problems are denoted as the problem of color constancy [321].

Above, we have introduced color as feature in visual information processing. Now, let us illustrate the influence of patterns or texture on the perceived color, where we define pixels as "any of the small discrete elements that together constitute an image (as on a television or computer screen) [181]". For example, a square perceived as orange can exist without orange pixels. It can be generated by regularly alternating red and yellow pixels (so called dithering), which create the orange percept, as is know from computer graphics and human perception studies. In contrast, from a physics/engineering point of view, the R, G, and B channel of a CRT tube tell us that 66.6% of the square is red and 33.3% is green. The latter example can be denoted as low level texture. At a higher level, texture is more clearly visible; e.g., a brick wall, a field of grass, wood, human skin. So, as Faugeras and Pratt already noted 25 years ago [83]: "The basic pattern and repetition frequency of a texture sample could be perceptually invisible, although quantitatively present." Whether or not a pattern should be analyzed as such depends on the application.

In computer vision, color and texture features are also utilized for shape extraction. Hereby, the representation of both features is of paramount importance. Until now, the results are promising but far from good. In addition, shape extraction is computationally expensive and, therefore, not usable for real time image processing and computer vision tasks. Humans are able to detect shape and process it with a high accuracy. In most cases, the human visual processing system works perfectly. However, it can be tricked. For example, artists such as Escher were able to deceive the human visual system.

Let me further illustrate the beauty of the human visual system. In 1658, Pascal [206] described that humans can see mirror symmetry at a glance [306, 310]. A broad range of fundamental research toward symmetry perception has been conducted; e.g., see Van der Helm and Leeuwenberg [110]. However, until now, no algorithm has been presented that completely describes the process of human symmetry recognition outside a controlled experimental setting; as Liu and Collins [162] stated: "choosing precisely which candidate is preferred by human perception is an open problem". As a consequence, the phenomenon of human symmetry recognition in natural object images (e.g., that contain a face), is hard if possible at all for computer vision.

In addition, humans can recognize objects that are occluded, without any problem. It is a fundamental issue in perception research. Thus far, the stimulus domain has been restricted to stylistic 2D line drawn stimuli [73, 150, 158]. Only in the last five years attempts have been made to extend occlusion research to the domain of natural object images. The ease with which humans recognize occluded objects is in sharp contrast with the problems such tasks reveal for computer vision.

So, despite the rapid evolvement of computer vision techniques, human vision is still superior in most aspects. Intrigued by the efficiency of human vision, on the one hand, we aim at adopting principles of human vision for computer vision and for CBIR techniques. On the other hand, image processing and CBIR techniques can and should be improved. So both from a computer vision and human vision point of view, CBIR should be approached.

1.2 Content-Based Image Retrieval (CBIR)

In 1992, Kato [139] introduced the term *content-based image retrieval* (CBIR), to describe his experiments on automatic retrieval of images from a database by color and shape features. Since then, CBIR arose as a new field of research.

CBIR is the application of computer vision to the image retrieval problem; i.e., the problem of searching for images in large image databases. Most image retrieval engines on the world wide web (WWW) make use of text-based image retrieval, in which images are retrieved based on their labels, descriptions, and surrounding text. Although text-based image retrieval is fast and reliable, it fully depends on the textual annotations that accompany images. Consequently, it requires every image in the database or on the WWW to be well annotated or labeled.

As Smeulders, Worring, Santini, Gupta, and Jain [270] noted in 2000 "CBIR is at the end of its early years" and is certainly not the answer to all problems. A quartet of arguments can be identified, which sustain the latter claim: (i) CBIR techniques still yield unacceptable retrieval results, (ii) they are restricted in the domain that is covered, (iii) they lack a suitable

user-interface and (iv) are mainly technology-driven and, subsequently, require the use of domain knowledge to fulfill their information need [235, 249].

In the last decade [6, 55, 122, 135, 184, 270, 308], a change in research perspective with respect to CBIR systems can be seen: from computer vision and pattern recognition to other disciplines such as cognitive science and psychology. Hence, the paramount importance to consider the human in the loop is more and more emphasized. Using knowledge about the user will provide insight in how the user-interface must be designed, how retrieval results may be presented, and it will categorize the typical information needs present with the general public [115]. Hence, in the line of research as discussed in this thesis the human is constantly in the loop of technological development.

1.3 Fields of application

Already a decade ago, Gudivada and Raghavan [100] identified twelve fields of application in which CBIR can prove its usefulness: crime prevention, the military, intellectual property, architectural and engineering design, fashion and interior design, journalism and advertising, medical diagnosis, geographical information and remote sensing systems, cultural heritage, education and training, home entertainment, and WWW searching. We will discuss three of these fields of application: (i) the WWW, (ii) professional databases in which dedicated CBIR techniques are applied, and (iii) photo books, as an application for customers.

1.3.1 The World Wide Web (WWW)

In 1945, Vannevar Bush described an application of electronics, which he named MEMEX [45, 197]. Bush envisioned how the user would be able to jump from one piece of information to another, facilitated by ingenious association and indexing techniques. This should result in a total experience of opinions and decisions of ourselves, of our friends, and of authorities. Within this, the concept of the WWW was already apparent. In 1966, Douglas Engelbart introduced hyper-text. With that he gave birth to Bush' dreams. Therefore, Engelbart is considered to be the founder of the Internet and the later arisen WWW [313]. In the early 1970s, as part of an Advanced Research Projects Agency (ARPA) research project on "internetworking", the Internet became truly operational.

The contrast is enormous between nowadays Internet and the text-based Internet as it was at its launch. The hyper-text protocol, as founded by Engelbart, can hardly satisfy the needs of Internet's current users. This is due to more and more digital multi modal information sources that are used; especially, images dominate the WWW with an average between 14.38 [189] and 21.04 [137] images per page. In principle, CBIR can be used to

retrieve these images from the WWW. However, CBIR on the (unrestricted) WWW suffers from time, computational, and storage (or space) complexity. A substantial effort has to be made before these are tackled.

1.3.2 Databases

Where the WWW is beyond the scope of current CBIR techniques, they can and are employed for databases. This is no different for the research presented in this thesis. Among other domains, the NWO ToKeN projects conduct research on the domains of cultural heritage and medical applications.

1.3.2.A Databases of museums

The Dutch Rijksmuseum states: "Research is the basic premise of all museum activity, from acquisition, conservation and restoration, to publication, education and presentation. In general, this research involves the object or work of art in the museum as being a source of information." [221] However, how can these sources of information be efficiently accessed and enhanced? The Rijksmuseum has made their collection accessible through a web-interface [222]. Their current interface provides the means to conduct 'classical' information retrieval; i.e., text-based search. Other recent initiatives are, for example, described in [67, 70, 98, 112, 236, 296].

In general, modern information retrieval techniques provide excellent results [113, 114] when two premises are satisfied: (i) a well annotated database is available and (ii) good choice of keywords is made, which both fits the query in mind and the keywords present in the database. In a professional environment, using a limited database, such an approach can be highly successful. In contrast, in most situations, no well annotated databases are present in an unrestricted domain, which are queried by non-professionals, using non-optimal keywords.

The general public does not know the style of a painter, the period he lived in, and how his paintings are named. Not seldomly, a visitor does not even know his name exactly. How to approximate a name, using keywords? It is possible, but by no means accessible to the general public. Within such a scenario, the user will not be to able access the data.

A professional will utilize his knowledge and query using his knowledge about the artist (e.g., name and country of residence), the object (e.g., title, material(s), technique(s)), the dates, the acquisition method, and possibly will be able to use his associations. Detailed queries can be defined resulting in retrieval results with a high precision. However, how to find objects that evoke the same atmosphere or trigger the same emotions? How to find objects with a similar expression although created using other techniques on differ-

ent materials? Systems that can answer such questions should be considered as being truly intelligent. Regrettably, such systems are far out of science's reach.

1.3.2.B Medical databases

Medical images have often been used for retrieval systems. Subsequently, the medical domain is often cited as one of the principal application domains for content-based access technologies. For example, in the radiology department of the University Hospital of Geneva alone, the number of images produced per day in 2002 was 12,000, and it is still rising."[185]

The rapid development in medical imaging is illustrated by the broad range of literature that has been published in the last decade [124, 149, 185, 256, 269, 278, 281, 301, 320]. Moreover, a still increasing number of medical image databases is available through websites [54, 271]. More and more, CBIR techniques are used to access these databases efficiently.

For a broad range of image types, in various medical domains, CBIR techniques are employed: dermatological images [241, 245], cytological specimens [179, 182], 3D cellular structures [11], histopathologic images [130], histology images [290], stenosis images (within cardiology) [198], MRIs (Magnetic Resonance Images) [103, 229], CT brain scans [160], ultrasound images [153], high resolution computed tomography (HRCT) scans of lungs [263], thorax radiographies [1], Functional PET (Photon Emission Tomography) images [47], and spine x-rays [5, 163]. Subsequently, a range of CBIR systems has been introduced, most of them dedicated to a specific type of medical images, see Table 1.1. These CBIR systems are already applied for professional databases. However, more general techniques should be developed and CBIR systems should be used more frequently, also outside the medical domain.

1.3.3 Photobook

More than a decade ago, one of the early CBIR systems was launched [209]: Photobook (see also [210]). Its name illustrates its intended domain of application: photo-collections. Nowadays, a still increasing, vast amount of people has a digital photo/video-camera. The ease of making digital photo's led to an explosion in digital image and video material. The exchange of these materials is facilitated through both Internet and mobile telephones. In addition, the costs for storing them have declined rapidly in the last years.

The exploded amount of digital image material is transported and stored on CDs, DVDs, and hard disks. In contrast, only a decade ago everybody used paper photo books to manage their photo-collection. The need emerged for digital photo books and, subsequently, a range of them was developed and computers and their operating systems were adapted to facilitate in handling (e.g., processing and browsing) multi-media information. However,

Table 1.1: An overview of various image types and the systems that are used to retrieve these images; adopted from [185].

Image types used	Names of the systems
HRCTs of the lung	ASSERT
Functional PET	FICBDS
Spine X-rays	CBIR2, MIRS
Pathologic images	IDEM, I-Browse, PathFinder, PathMaster
CTs of the head	MIMS
Mammographies	APKS
Images from biology	BioImage, BIRN
Dermatology	MELDOQ, MEDS
Breast cancer biopsies	BASS
Varied images	I2C, IRMA, KMed, COBRA, MedGIFT, ImageEngine

where in paper photo books people wrote small texts and placed dates accompanying photos, in their digital counterparts this effort is not made. Not in the last place, this will be due to the vast amount of digital images compared to analog ones, made in the recent past. The latter is due to the fact that (i) people tend to take an image without hesitating and (ii) there are no developing and publishing costs. As a consequence, the amount of digital photo's in private photo collections is much larger than with their analog counterparts.

Since the digital image collections are not or poorly annotated, text-based image retrieval cannot be applied. Manual searching is a frequently used alternative but becomes less attractive with the increasing size of the private, digital photo collections. In "How do people organize their photographs?", Rodden [231] describes a research with the aim "to gain some insight into how computer-based systems to help people organize these images might be designed." In the years before and after Rodden's suggestions, a few methods for browsing through photo collections have been proposed. Already in 1994, Gorkani and Picard [97] proposed to utilize "Texture Orientation for Sorting Photos at a Glance". In 1997, Ma and Manjunath [168] launched "NeTra: a toolbox for navigating large image databases" (see also [169]). Shneiderman and Kang [262] proposed "Direct Annotation: A drag-and-drop strategy for labeling photos". Despite the effort made, no full-grown solution has been introduced; so, the need for the development of adequate CBIR techniques was stressed.

The WWW and large professional databases (to a lower extent) suffer from a computational burden, due to the large amount of (high quality) image material. No such problems are present with private image collections. Where private image collections can be too large for manual searching, they are small compared to most professional databases. So, CBIR systems can provide a substantial contribution in managing private photo collections.

1.4 The envisioned approach

CBIR is mostly approached by experts in the field of image processing. More recently, experts in cognitive science and artificial intelligence conducted research toward CBIR systems. The research described in this thesis envisioned a multi-disciplinary approach in which methods and techniques from both worlds of science are united.

The research presented in this thesis started with fundamental research: categorization of color stimuli by participants in an experimental setting. Statistics, as frequently used in social sciences were used to analyze the results. The analysis of the experimental data continued with a range of techniques as used in artificial intelligence and image processing. A new texture analysis technique will be introduced utilizing human color processing scheme and combining two image processing schemes. Next, human and artificial texture classification were compared to each other.

In the last phase of this project, color and texture analysis were exploited for image segmentation and shape extraction. All techniques were combined in one efficient human-centered object-based image retrieval engine.

During several phases in the research, techniques and methods were evaluated within a newly developed CBIR-benchmark. These evaluations were conducted by humans; hence, the humans were almost continuously in the loop of the development of the CBIR techniques.

1.5 Outline of the thesis

In the previous pages, we provided a broad overview of central concepts and introduced the perspective from which this research was done. The research itself, followed by a general discussion, is presented in the remaining part of this thesis. It comprises a range of research methods; e.g., fundamental research, image processing techniques, evaluation of techniques, methods, and algorithms. These methods are used in five lines of research: (i) the 11 color categories demystified, (ii) benchmarking techniques, (iii) texture analysis, (iv) image segmentation and shape extraction, and (v) techniques for Euclidean distance transforms. Across these lines of research, the first line of research: the 11 color categories, as used by humans in processing color, are the foundation of the image analyzes schemes developed. We will now provide a brief overview of the five lines of research and subsequently, of the chapters in which they are presented, starting with the 11 color categories.

1.5.1 Research line 1: 11 Color categories

Most research that will be presented in this thesis, is related to color. Therefore, we will start with discussing this feature in Chapter 2. In the first part of this chapter, color histograms, color quantization, and color spaces are discussed. In the second part of this chapter, an overview of another view on color: 11 color categories is provided.

In Chapter 3, a questionnaire and two experiments that sustained the use of 11 color categories in a computer environment are discussed. From the experimental data, Color Look-Up Table (CLUT) markers came forth, which can be considered useful for modeling human color categorization. The complete table of CLUT markers is available as Appendix A. In Chapter 4, image processing techniques are introduced, which are applied on the CLUT markers. The latter two chapters provide the ingredients for an efficient human-centered color space segmentation, as described in Chapter 5.

1.5.2 Research line 2: Benchmarking Content-Based Image Retrieval techniques

Since the aim of this project was to develop intelligent CBIR techniques, we choose to validate the color space segmentation, as described in Chapter 5, in such a setting. For this purpose, a CBIR benchmark was developed, as is described in Chapter 6. Moreover, the benchmark provided the means to take the human in the loop of development.

In "The utilization of human color categorization for content-based image retrieval" (Chapter 7), the CBIR benchmark is applied: seven different CBIR engines were tested, each defined by a combination of a color quantization scheme (or color categorization) and a distance measure. In Chapter 8: "Utilizing color categories and color distributions", we present a second CBIR benchmark. A large group of users participated in this benchmark. A new distance measure, based on the 11 color categories quantization scheme, was introduced, providing additional color information.

1.5.3 Research line 3: Texture

In the third line of research, texture analysis methods are studied. Most texture description methods are designed for the Intensity (gray value) domain of images. An overview of Intensity-based texture descriptors is provided in Chapter 9.

With the rise of color photo's and color television, in the second half of the 20^{th} century, the interest in colorful texture analysis grew. However, in practice most texture analysis methods were still developed for the Intensity dimension only. This was due to the computational burden when analyzing image material in a 3D color space and due to the

complexity of the phenomenon color. In the last decade, a few color-based texture analysis methods were developed; Chapter 9 provides an evaluation of them.

In Chapter 10: "Parallel-sequential texture analysis", the feasibility of image classification by way of texture was explored. Hereby, a range of color quantization schemes and color spaces were utilized. Moreover, the parallel-sequential texture analysis was launched, which combines global color analysis with texture analysis, based on color.

As was indicated in Section 1.1, human vision can be taken as inspiration for CBIR/computer vision techniques. In Chapter 11, a study is discussed in which an attempt is made to measure up to human texture classification. In this research, we compare artificial texture analysis and classification techniques with human texture classification.

1.5.4 Research line 4: Image segmentation and shape extraction

For image segmentation, frequently texture is utilized. Using local texture and color features of image material, an image can be segmented. If needed, the shape of objects can be approximated, using the segments found.

The development of a human-centered object-based image retrieval engine is discussed in Chapter 12. Coarse color image segmentation was applied, using the agglomerative merging algorithm. The CBIR benchmark, as introduced in Chapter 6, utilizing the intersection distance measure, was used to evaluate the engines developed. Based on 15 features (i.e., the 11 color categories and 4 texture features), the center objects of images were analyzed and images with comparable objects were retrieved. In "Human-centered object-based image retrieval" (Chapter 13), the coarse image segmentation techniques of Chapter 12 were adapted for shape extraction. Using pixelwise classification based on the 11 color categories, followed by smoothing operations, the shapes of objects were extracted. Next, the Vind(X) shape matching algorithm was applied for shape matching. Four CBIR engines were applied on the same data-set, exploiting: (i) color and texture of the object versus complete images, (ii) color and texture of the object, (iii) shape, and (iv) color, texture, and shape combined.

1.5.5 Research line 5: Euclidean distance transforms

This research line concerns the field of computational geometry. Within this research line, the Fast Exact Euclidean Distance (FEED) transform was launched. This line of research ran in parallel with the other research lines. In Chapter 4, distance mapping is discussed in general and FEED more specific, for binary as well as for multiple class data (such as the 11 color categories). In Chapter 5, FEED was applied in order to bridge the gap between

(limited) experimental data concerning the 11 color categories and a quantization of color space based on it (cf. Chapter 3 and 5), such that it can be applied for CBIR purposes.

The paper in which FEED was initially launched, can be found in Appendix C. A substantially faster parallel (timed) implementation (tFEED) was introduced a few months later, see Schouten, Kuppens, and Van den Broek [253]. A dimension independent description of the FEED algorithm is defined in Schouten, Kuppens, and Van den Broek [251], accompanied with the launch of an implementation of FEED for 3D data: 3D-FEED.

2

Color representation

Abstract

In this chapter, general color concepts are introduced, which will be used in the remainder of this thesis, for color analysis, colorful texture analysis, shape extraction, and CBIR. First, the color histogram is described: a discrete function that quantizes the distribution of colors of an image. Next, several color spaces and their color quantization schemes are discussed. In addition, an alternative view on color is presented: 11 color categories. A brief history is sketched of a century of research on these color categories. Consecutively, the Sapir-Whorf hypothesis, the work of Brown and Lenneberg, and the theory of Berlin and Kay, are discussed.

Color is the sensation caused by light as it interacts with our eyes and brain. The perception of color is greatly influenced by nearby colors in the visual scene. The human eye contains two types of visual receptors: rods and cones. The rods are responsive to faint light and therefore, sensitive to small variations in luminance. The cones are more active in bright light and are responsible for color vision. Cones in the human eye can be divided in three categories, sensitive to long, middle, and short wavelength stimuli. Roughly these divisions give use to the sensations of red, green, and blue.

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that facilitates object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray [95].

In this chapter, general color concepts, as used in this thesis, will be introduced. We will start with a description of the color histogram. Next, color quantization will be explained, followed by the description of several color spaces and their quantization methods. In addition, the research conducted in the last century toward an alternative view on color is presented: 11 color categories. We end this chapter, with the introduction of the distance measures that have been applied.

2.1 The color histogram

The color histogram is a method for describing the color content of an image, it counts the number of occurrences of each color in an image [321]. The color histogram of an image is rotation, translation, and scale-invariant; therefore, it is very suitable for color-based CBIR: content-based image retrieval using solely global color features of images. However, the main drawback of using the color histogram for CBIR is that it only uses color information, texture and shape-properties are not taken into account. This may lead to unexpected errors; for example, a CBIR engine using the color histogram as a feature is not able to distinguish between a red cup, a red plate, a red flower, and a red car as is illustrated in Figure 2.1.

Many alternative methods have been proposed in the literature. They include color moments [217, 285], color constants [85, 321], color signatures [142], color tuple histograms [104], color coherent vectors [105], color correlograms [88], local color regions [142], and blobs [51]. These methods are concerned with optimizing color matching techniques on a spatial level; i.e., utilizing the spatial relations between pixels, in relation to their colors. However, they disregard the basic issue of intuitive color coding. In other words, the way the engine is processing color, is not related to human color processing. In our opinion, prior to exploring these techniques, the issue of color coding (or categorization) should be stressed; e.g., as can be done with the color histogram.

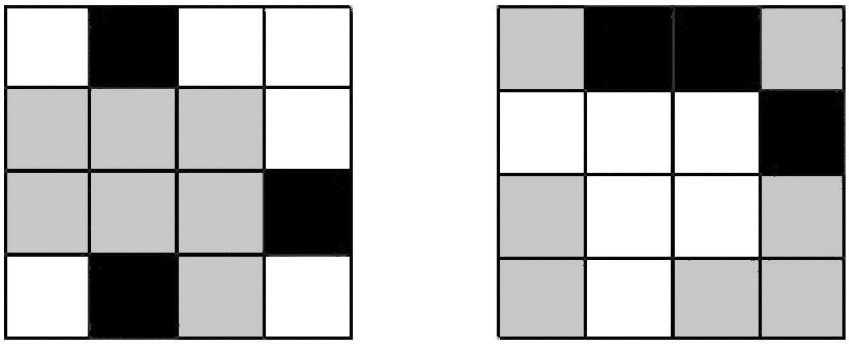


Figure 2.1: Two distinct images are shown. However, when represented by their color histograms, they are judged as identical.

2.2 Color quantization

In order to produce color histograms, color quantization has to be applied. Color quantization is the process of reducing the number of colors used to represent an image. A quantization scheme is determined by the color space and the segmentation (i.e., split up) of the color space used. A color space (see Section 2.3) is the representation of color. Typically (but not necessarily), color spaces have three dimensions and consequently, colors are denoted as tuples of (typically three) numbers.

In applying a standard quantization scheme on a color space, each axis is divided into a number of parts. When the axis are divided in k, l, and m parts, the number of colors (n) used to represent an image will be $n = k \cdot l \cdot m$. A quantization of color space in n colors is often referred to as a n-bins quantization scheme. Figure 2.2 illustrates the effect of quantizing color images. The segmentation of each axis depends on the color space used. In the next section, different color spaces and their quantization methods will be described.

(a) (b) (c)

Figure 2.2: (a) The original image using 256^3 colors, (b) quantized in 8 bins, and (c) quantized in 64 bins, using RGB color space. See Figure B.1 in Appendix B for large color versions of these three photos.

2.3 Color spaces

A color space specifies colors as tuples of (typically three) numbers, according to certain specifications. Color spaces lend themselves to (in principle) reproducible representations of color, particularly in digital representations, such as digital printing or digital electronic display. The purpose of a color space is to facilitate the specification of colors in some standard, generally accepted way [95].

One can describe color spaces using the notion of perceptual uniformity [321]. Perceptually uniform means that two colors that are equally distant in the color space are perceptually equally distant. Perceptual uniformity is a very important notion when a color space is quantized. When a color space is perceptually uniform, there is less chance that the difference in color value due to the quantization will be noticeable on a display or on a hard copy.

In the remainder of this section, several color spaces with their quantization schemes will be described. In addition, the conversion of color images to gray-scale images, using the specific color space, will be described. The quantization of color images transformed into gray-scale images, is independent of the color spaces: the gray-scale axis is divided in the number of bins needed for the specific quantization scheme. In this thesis, gray-scale images were quantized in 8, 16, 32, 64, and 128 bins.

2.3.1 The RGB color space

The RGB color space is the most used color space for computer graphics. Note that R, G, and B stand here for intensities of the Red, Green, and Blue guns in a CRT, not for primaries as meant in the CIE [59] RGB space. It is an additive color space: red, green, and blue light are combined to create other colors. It is not perceptually uniform. The RGB color space can be visualized as a cube, as illustrated in Figure 2.3.

Each color-axis (R, G, and B) is equally important. Therefore, each axis should be quantized with the same precision. So, when the RGB color space is quantized, the number of bins should always be a cube of an integer. In this thesis, 8 (2^3), 64 (4^3), 216 (6^3), 512 (8^3), and 4096 (16^3) bins are used in quantizing the RGB color space. The conversion from a RGB image to a gray value image simply takes the sum of the R, G, and B values and divides the result by three.

2.3.2 The HSx color spaces

The HSI, HSV, HSB, and HLS color spaces (conventionally called 'HSx') are more closely related to human color perception than the RGB color space [159], but are still not perceptually

uniform.

The axes from the HSx color spaces represent hue, saturation, and lightness (also called value, brightness and intensity) color characteristics. The difference between the different HSx color spaces is their transformation from the RGB color space. They are usually represented by different shapes (e.g., cone, cylinder). In Figure 2.3, the HSV color space is visualized as a cone.

Hue is the color component of the HSx color spaces. Hue is an angle between a reference line and the color point in RGB space [53], the range of this value is between $0°$ and $360°$, for example blue is $240°$. According to the CIE (Commission Internationale de lÉclairage) [59], hue is *"the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors, red, yellow, green, and blue, or a combination of two of them"*. In other words, hue is the color type, such as red or green. Also according to CIE, saturation is *"the colorfulness of an area judged in proportion to its brightness"*. In the cone, the saturation is the distance from the center of a circular cross-section of the cone, the 'height' where this cross-section is taken is determined by the Value, which is the distance from the pointed end of the cone. The value is the brightness or luminance of a color, this is defined by CIE as *"the attribute of a visual sensation according to which an area appears to emit more or less light"*. When Saturation is set to 0, Hue is undefined. The Value-axis represents the gray-scale image.

The HSV color space can easily be quantized, the hue is the most significant characteristic of color so this component gets the most fine quantization. In the hue circle, the primary colors red, green, and blue, are separated by $120°$. The secondary colors, yellow, magenta, and cyan, are also separated by $120°$ and are $60°$ away from the two nearest primary colors.

The most common quantization of the HSV color space is in 162 bins, where hue gets 18 bins and saturation and value both get 3 bins. When hue is divided in 18 bins, each primary color and secondary color is represented with three subdivisions. In this thesis, the HSV color space is quantized in 27 ($3 \times 3 \times 3$), 54 ($6 \times 3 \times 3$), 108 ($12 \times 3 \times 3$), 162 ($18 \times 3 \times 3$), and 324 ($36 \times 3 \times 3$) bins.

2.3.3 The YUV and YIQ color spaces

The YUV and YIQ color spaces are developed for television broadcasting. The YIQ color space is the same as the YUV color space, where the I-Q plane is a $33°$ rotation of the U-V plane. The Y signal represents the luminance of a pixel and is the only channel used in black and white television. The U and V for YUV and I and Q for YIQ are the chromatic components.

The Y channel is defined by the weighted energy values of R(0.299), G(0.587), and B(0.144) . The YUV and YIQ color spaces are not perceptually uniform. When the YUV

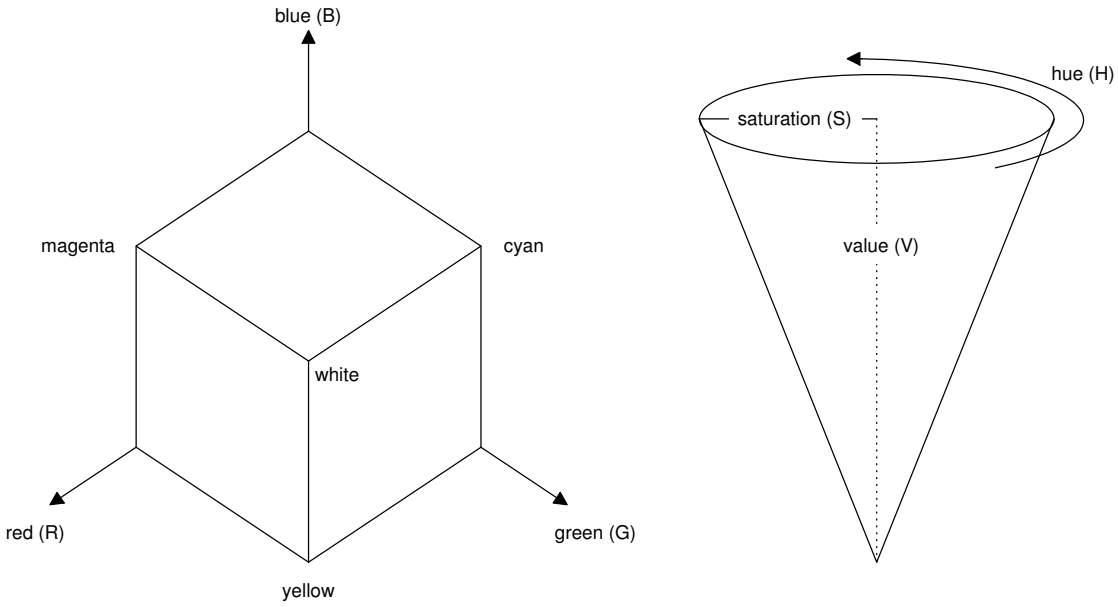


Figure 2.3: The RGB (red, green, and blue) and HSV/HSI (hue, saturation, and value/intensity) color space.

and YIQ color spaces are quantized, each axis is quantized with the same precision. The quantization schemes used for the YUV and YIQ color spaces in this theses are: 8 (2^3), 27 (3^3), 64 (4^3), 125 (5^3), and 216 (6^3) bins.

To optimize color appearance the YUV color space is often sampled. The samplings we used to construct the color correlogram are: 4:4:4, 4:2:2, and 4:1:1, where the numbers denote the relative amount of respectively Y on each row, U and V on each even-numbered row, and U and V on each odd-numbered row in the image.

2.3.4 The CIE XYZ and LUV color spaces

The first color space developed by the CIE is the XYZ color space. The Y component is the luminance component defined by the weighted sums of R(0.212671), G(0.715160), and B(0.072169). The X and Z are the chromatic components. The XYZ color space is perceptually not uniform. In quantizing the XYZ space, each axis is quantized with the same precision.

The CIE LUV color space is a projective transformation of the XYZ color space that is perceptually uniform. The L-channel of the LUV color space is the luminance of the color. The U and V channels are the chromatic components. So, when U, and V are set to 0, the L-channel represents a gray-scale image.

In quantizing the LUV space, each axis is quantized with the same precision. For both the XYZ color space and the LUV color space, the following quantization schemes are used: 8 (2^3), 27 (3^3), 64 (4^3), 125 (5^3), and 216 (6^3) bins.

2.3.5 Toward another form of color representation

So far, we have discussed the color spaces used in our research and their quantization schemes. All research in the field of CBIR and computer vision relies on the quantization of color spaces. However, in this thesis we will introduce another approach toward color space quantization (or segmentation); see also Chapter 5.

In the next section, we will provide a rapid guide through half a century of research on human color categories, a concept that originates from anthropology and is discussed in the field linguistics and psychology. In contrast, it was mainly ignored in the field of CBIR and computer vision. In this thesis, a full color space segmentation is presented, which is the foundation for all possible CBIR techniques, as utilized through this thesis.

2.4 Color categories: A century of research

In this part, we provide an overview of a century of research concerning human color categories. We start with the general Sapir-Whorf hypothesis in Section 2.4.1. Second, we will discuss the work of Brown and Lenneberg, who studied the Sapir-Whorf hypothesis with respect to human color coding (see Section 2.4.2). Last, in Section 2.4.3, we discuss the thorough research of Berlin and Kay, as reported in 1969. Each section concludes with the discussion of recent research, including pros and cons.

2.4.1 The Sapir-Whorf hypothesis

Sapir was inspired by the work of Wilhelm von Humboldt [120], who stated: "Man lives in the world about him principally, indeed exclusively, as language presents it to him." According to Humboldt languages differ from one another; thought and language are inseparable; and, therefore, each speech community embodies a distinct world-view.

In 1929, Edward Sapir stated in his "The Status of Linguistics as a Science" [240]: "Human beings do not live in the objective world alone, nor alone in the world of social activity as ordinarily understood, but are very much at the mercy of the particular language which has become the medium of expression in their society." Further, he stated: "The fact of the matter is that the 'real world' is to a large extent unconsciously built up on the language habits of the group."

In addition, Benjamin Lee Whorf (1940/1956) stated in his "Science and Linguistics" [316, 317]: "We cut nature up, organize it into concepts, and ascribe significances as we do, largely because we are parties to an agreement to organize it in this way - an agreement that holds throughout our speech community and is codified in the patterns of our language.

The agreement is, of course, an implicit and unstated one, but its terms are absolutely obligatory."

One can distinguish two theories, concerning linguistic determinism, in their writings: (i) the language we speak determines the way we interpret the world around us [240] and (ii) a weaker theory, which states that language influences our representation of the world [316, 317].

However, neither Sapir nor Whorf formally described their theory nor supported it with empirical evidence. Nevertheless, in the 1930s and 1940s, the Sapir-Whorf hypothesis has caused controversy and spawned research in a variety of disciplines (e.g., linguistics, psychology, philosophy, and anthropology).

By dovetailing the Sapir-Whorf hypothesis, Lucy [166, 167] and Slobin [268] have demonstrated that language can directly influence our thoughts. Through verbal limitation, grammatical focus, and structural emphasis, oral communication can pattern our very way of thinking. Cultural anthropologist Andy Clark concludes that language not only "confers on us added powers of communication; it also enables us to reshape a variety of difficult but important tasks into formats suited to the basic computational capacities of the human brain" [57]. Hence, cultures with different structural axioms result in different computational capacities.

However, through the years, more studies appeared that dispute the Sapir-Whorf hypothesis. For example, Osgood [204] found that "human beings the world over, no matter what their language or culture, do share a common meaning system, do organize experience along similar symbolic dimensions." Similar conclusions were drawn by Schlesinger [244].

In the current section, the influence of language on cognition and, more specific, perception was discussed. This was done on a rather abstract level. In the next section, the relation between language and the perception of colors will be discussed. With the description of this relation, an important concept is introduced that can be considered as the foundation for the present thesis.

2.4.2 Human color coding

In 1954, Roger W. Brown and Eric H. Lenneberg [43] reported their influential "a study in language and cognition". They stated "that more nameable categories are nearer the top of the cognitive 'deck' [43]".

Brown and Lenneberg [43] introduced their experiment with: "Sensory psychologists have described the world of color with a solid using three psychological dimensions: hue, brightness, and saturation. The color solid is divisible into millions of just noticeable differences; Science of Color [202] estimates 7,500,000. The largest collection [82, 173] of English

color names runs less than 4,000 entries, and of these only 8 occur very commonly [295] *(i.e., red, orange, yellow, green, purple, pink, and brown)*[1]. Evidently there is considerable categorization of colors. It seems likely to us that all human beings with normal vision will be able to make approximately the same set of discriminations. This ability appears to depend on the visual system, which is standard equipment for the species. Whatever individual differences do exist are probably not related to culture, linguistic or extra linguistic. It does not follow that people everywhere either see or think of the world in the same way. Cultural differences probably operate on the level of categorization rather than controlled laboratory discrimination."

They found that English color naming and color recognition are related [43]. Brown and Lenneberg [43] "guess it is that in the history of a culture the peculiar features of the language and thought of people probably develop together." They continue (and conclude) with: "In the history of an individual born into a linguistic community the story is quite different. The patterned responses are all about him. They exist before he has the cognitive structure that will enable him to pattern his behavior in the approved fashion. Simple exposure to speech will not shape anyone's mind. To the degree that the unacculturated individual is motivated to learn the language of a community, to the degree that he uses its structure as a guide to reality, language can assume a formative role. [43]"

Brown and Lenneberg's statements can be explained in favor of both the linguistic relativism [244] and the cognitive universalism [208]. Research of Lucy and Shweder [165] and Kay and Kempton [141] provided support for the linguistic relativity hypothesis with respect to color memory. In contrast, based on his cross-cultural color sorting test, Davies concluded strong cognitive universalism [71].

2.4.3 Berlin and Kay's Basic color terms

We will now discuss the research of Brent Berlin and Paul Kay's book as described in "Basic color terms: Their universality and evolution" [16]. Their research indicates that "semantic universals do exist in the domain of color vocabulary. Moreover, these universals appear to be related to the historical development of all languages in a way that can properly be termed evolutionary. [16]" This statement was based on the data gathered from native-speaking informants of twenty languages, from unrelated language families [16].

It appeared that "although different languages encode in their vocabularies different numbers of color categories, a total universal inventory of exactly eleven basic color categories exist from which the eleven or fewer basic color terms of an given language are always drawn." Berlin and Kay also found that "the distributional restrictions of color terms across languages are:

[1] italics are added by the author

1. All languages contain terms for white and black.
2. If a language contains three terms, then it contains a term for red.
3. If a language contains four terms, then it contains a term for either green or yellow (but not both).
4. If a language contains five terms, then it contains terms for both green and yellow.
5. If a language contains six terms, then it contains a term for blue
6. If a language contains seven terms, then it contains a term for brown
7. If a language contains eight or more terms, then it contains a term for purple, pink, orange, gray, or some combination of these."

"Further evidence for the cross-language universality of color foci is that the location of color foci varies no more between speakers of different languages than between speakers of the same language. [16]" Let us further adopt from Berlin and Kay [16] that "whenever we speak of color categories, we refer to the foci of categories, rather than to their boundaries or total area, except when stating otherwise."

Independent of the cause of the universal color categories, most researchers confirm their existence. However, recently Roberson et al. [226, 227, 228] provided evidence that supports that color categories are not universal. Others such as Benson [10] simply stated: "Color categories make the world easier to live in. Granny Smith (green) and Red Delicious (red) apples belong in different categories (bins)." The majority of research reported follows Benson (e.g., [75, 219]). Let us, therefore, adopt the opinion that color categories are, at least almost, universal. This, due to either linguistic relativism, through cognitive universalism, or through a combination of both hypotheses.

2.4.4 Quantization of the 11 color categories

The 11 color categories can be quantized into more color categories by dividing the Hue, Saturation, and Intensity values after the 11 color segmentation is complete. For example, a 70 bins quantization can be accomplished by dividing Hue, Saturation, and Intensities in 2 parts. If all 11 categories were colorful categories, the number of bins would be 88 ($2 \times 2 \times 2 \times 11$). However, three of the color categories are gray-scales (i.e., black, white, and gray) and subsequently, Saturation is 0 and Hue is undefined for these categories (see Section 2.3.2). Therefore, for these three categories, dividing each axes in 2 bins (or segments) does not result in 24 (3×8) colors but in 6 (3×2) colors (gray-scales). The number of categories is thus determined as follows: $88 - 24 + 6 = 70$. In the remainder of this thesis, the 11 color categories were quantized in 11, 27 ($11 \times (3 \times 1 \times 1) - 6$), 36 ($11 \times (4 \times 1 \times 1) - 8$), 70 ($11 \times (2 \times 2 \times 2) - 24 + 6$), and 225 ($11 \times (3 \times 3 \times 3) - 81 + 9$) bins.

2.5 Discussion

In this chapter, the basic color terms used in this thesis are explained. Moreover, an alternative view on color analysis has been introduced: 11 color categories. In the next chapter, two experiments are discussed in which the 216 W3C web-safe colors were assigned to the 11 color categories. The resulting data are the foundation of a new, human-based color space segmentation, as will be introduced in Chapter 5.

The quantizations as discussed in this chapter, will be utilized for texture analysis purposes, as will be shown in the second half of this thesis, starting with Chapter 9. In addition, the 11 color categories are utilized for segmentation of image material (see Chapter 12) and for shape matching purposes (see Chapter 13).

3

Modeling human color categorization: Color discrimination and color memory

Abstract

In Content-Based Image Retrieval, selection based on color is done using a color space and measuring distances between colors. Such an approach yields non-intuitive results for the user. We introduce color categories (or focal colors), determine that they are valid, and use them in two experiments. The experiments conducted, reveal a difference between color categorization of the cognitive processes color discrimination and color memory. In addition, they yield a Color Look-Up Table, which can improve color matching, that can be seen as a model for human color matching.

This chapter is an adapted version of:
Broek, E. L. van den, Hendriks, M. A., Puts, M. J. H., and Vuurpijl, L. G. (2003). Modeling human color categorization: Color discrimination and color memory. In T. Heskes, P. Lucas, L. G. Vuurpijl, and W. Wiegerinck (Eds.), *Proceedings of the Fifteenth Belgium-Netherlands Artificial Intelligence Conference (BNAIC2003)*, p. 59-66. October 23-24, The Netherlands - Nijmegen.

3.1 Introduction

The origin of the color *lilac* lays in the Sanskrit *nilla 'dark blue'*, of which the Persian made *nIlak 'bluish'*, from *nIl 'blue'*. In the Arabic the meaning evolved to a description of a plant with flowers of this color: the *Sering*. In 1560 the *Sering* was brought to Vienna, by an Austrian ambassador. From there the plant reached France. There the word's meaning evolved to *"a variable color averaging a moderate purple"* [181, 288].

So, there is more with colors than one would think at a first glance. The influence of color in our everyday life and the ease with which humans use color are in strong contrast with the complexity of the phenomenon color (topic of research in numerous fields of science; e.g., physics, biology, psychology, computer vision).

In this chapter, we focus on the use of colors in the field of *Content-Based Image Retrieval (CBIR)* [235, 270]. On the one hand, one has to take into account the RGB-color space used by the computer, the environmental conditions, etc. On the other hand, human color perception is of utmost importance. Since (human) users judge the retrieval results, the CBIR's matching algorithms need to provide a match that the user can accept. The complexity of this constraint is illustrated by the amount of available color spaces, such as: RGB, HSV, CIE [59] XYZ, and Munsell [230] [79]. However, none of these color spaces models human color perception adequately.

In our opinion, one should consider color in CBIR from another perspective; i.e., that of the focal colors or color categories (i.e., black, white, red, green, yellow, blue, brown, purple, pink, orange, and gray; see also Chapter 2). People use these categories when thinking, speaking, and remembering colors. Research from diverse fields of science emphasize the importance of them in human color perception. The use of this knowledge can possibly provide a solution for the problems of color matching in CBIR.

Most CBIR-engines distinguish two forms of querying, in which the user uses either an example image (*query-by-example*) or defines features by heart, such as: shape, color, texture, and spatial characteristics (query-by-content). In the latter case, we are especially interested in *query-by-color*. At the foundation of each of these queries lies a cognitive process, respectively color discrimination and color memory. Let us illustrate the importance of the distinction between *query-by-example* and *query-by-color* by a simple example. Imagine you want to find images of brown horses. In the case of *query-by-example*, the resulting images will be matched on the example image: a process of color discrimination is triggered. In this process, the colors are (directly) compared to each other. In the case of *query-by-color*, we need to try to imagine the color brown. Probably, you will not have a clear color in mind, but a fuzzy idea or a fuzzy set of colors: a *color category*, based on your *color memory*. All individual elements of this brown set (or category) are acceptable colors. There is no need for several types of brown. Providing the keyword 'brown' or pressing a button resembling

the fuzzy set brown is sufficient.

In both forms of querying the CBIR-system can use a *Color Look-Up Table (CLUT)* for the determination of the elements of this set, described by R, G, and B-values. The set is fuzzy due to the several influences on the color (of the object of interest), such as the color of the surrounding and the semantic context in which the object is present.

However, it is clear that a distinction should be made between color categorization by discrimination and color categorization by memory. An important distinction because humans are capable of discriminating millions of colors but when asked to categorize them by memory, they use a small set colors: *focal colors* or *color categories* [16, 42, 93, 232]. Despite the fact that the importance of such a distinction is evident, this differentiation is not made in CBIR-systems.

In the remainder of this chapter, a question is posed and two experiments will be executed. The question posed to the subjects is: "Please write down the first 10 colors that come to mind." With the experiments we show the difference between color categorization by color discrimination and by color memory. Hence, this research will show that:

- The use of *color categories* is valid in a CBIR context,
- The RGB-color space can be assigned to *color categories*,
- There is a difference in color categorization using color discrimination or color memory.

Moreover, we will present markers, by which the color space is divided, on which a *CLUT* for CBIR can be employed. With that a new model of human color categorization is introduced.

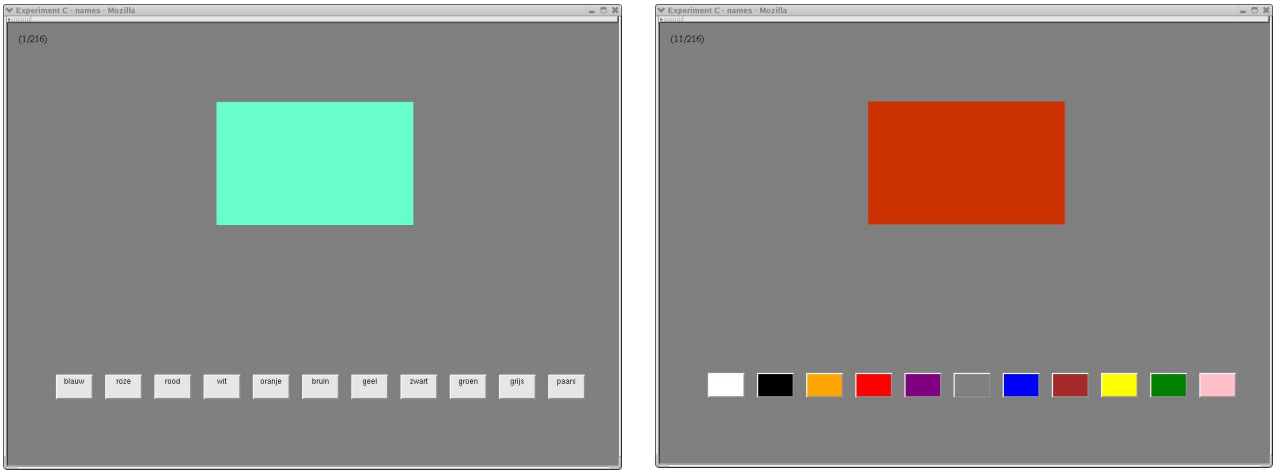
3.2 Method

3.2.1 Subjects

Twenty-six subjects with normal or corrected-to-normal vision and no color deficiencies, participated. They participated either voluntary or within the scope of a course. The first group were employees and the latter were students of the Radboud University Nijmegen. They were naive as to the exact purpose of the experiment.

3.2.2 Equipment

An attempt was made to create an *average office environment*. Stimuli were presented on a 17" CRT monitor (ELO Touchsystems Inc., model: ET1725C), with a resolution of 1024 x 768



(a) (b)

Figure 3.1: Screendumps of the user interfaces of (a) the color memory experiment, with gray, labeled buttons, and of (b) the color discrimination experiment with unlabeled, colored buttons. So, the color classification had to be employed based on respectively color memory color discrimination. See Figures B.2 and B.3 in Appendix B for large color versions of these screendumps.

pixels at a refresh-rate of 75Hz. The experiment was conducted in an room with average office lighting: a Cool White Fluorescent light source: TL84 was present, its color temperature: 4100K (Narrow Band Fluorescent), as used primarily in European and Asian office lighting.

The experiments ran on a PC with an Intel Pentium II 450 MHz processor, 128mb RAM, a Matrox Millennium G200 AGP card, and with a Logitech 3-button Mouseman (model: M-S43) as pointing-device. The experiments were conducted in a browser-environment with Internet Explorer 6.0 as browser and Windows 98SE as operating system, using 16-bit colors, where respectively 5, 6, and 5 bits are assigned to the red, green, and blue channel.

3.2.3 Stimuli

The stimuli were the full set of *the 216 web-safe colors*[1]. These are defined as follows: The R, G, and B dimensions (coordinates) are treated equally. Their minimum value is 0, the maximum value of each of the dimensions is 255. For each dimension 6 values are chosen on equal distance, starting with 0. So, for the RGB-values 0 (0%), 51 (20%), 102 (40%), 153 (60%), 204 (80%), and 255 (100%) are chosen. Each of these 6 values is combined with each of the 6 values of the 2 other dimensions. This results in $6^3 (= 216)$ triple of coordinates in the RGB-space. These RGB-values result for both Internet Explorer and Netscape under both the Windows and the Mac operating system, in the same (non-dithered) colors iff the operating system uses at least 8-bit (256) colors.

[1] http://www.vu.msu.edu/pearls/color/1.htm

The stimulus (width 9.5 cm and height 6.0 cm) was presented in the center of the screen, on a gray background. Below the stimulus 11 buttons were placed (width: 1.8 cm and height 1.2 cm; width between: 0.6 cm) (see Figures 3.1a and 3.1b). In the color memory experiment the buttons were labeled with the names of the 11 focal colors (see Figure 3.1a); in the color discrimination experiment each of the buttons did have one of the 11 focal colors (see Figure 3.1b). The 11 *focal colors* were presented conform the sRGB standard of the World Wide Web consortium (W3C) [284]. The button of choice was selected with one mouse click upon it.

3.2.4 Design

Half of the participants started with the color discrimination experiment, the other half started with the color memory experiment. Each experiment consisted of four blocks of repetitions of all stimuli (in a different order), preceded by a practice session. Each block consisted of the same 216 stimuli, randomized for each block and for each participant. In addition, the 11 buttons were also randomized for each block and for each participant. The practice session consisted of 10 stimuli. Block, stimulus, and button order was the same for both experiments. Between the stimuli a blank screen was provided for one second, with a gray color. Table 3.1 provides a schematic overview of this design.

The participants were asked to take a short break between the blocks of repetition, within each experiment and to take a somewhat longer break between both experiments. The duration of the breaks was determined by the subjects. In total, a complete session took on the average 70 minutes, including breaks.

3.2.5 Procedure

The global scope of the experiment was explained. After that a small questionnaire was completed. The first task was to write down the 10 colors that arise from memory first. Next, the design of the experiments was explained. The subjects were instructed for the color memory experiment to categorize the stimulus into one of the color categories, represented by their names. In the color discrimination experiment, the subjects were asked to choose

Table 3.1: The design of the experimental session: a practice session followed by the color memory and the color discrimination experiment, each consisting of four blocks, with as stimuli the randomized 216 W3C web-safe colors. The participants were able to take a break, as denoted by 'b' and ' **b** ', for respectively a short and longer break.

practice	**b**	experiment 1							**b**	experiment 2						
		1	b	2	b	3	b	4		1	b	2	b	3	b	4

one of the 11 focal-colors that best resembled the stimulus. Last, it was emphasized that there were no wrong answers and that if questions would arise they could be asked during one of the breaks.

3.3 Results

3.3.1 Mentioning of color names

For the determination of the confidence intervals we have used the modified Wald method [2] that proved to work well with a limited number of measurements and with proportions close to 0 or 1.0; both the case in the present research. The proportion or frequency of appearance was determined by:

$$p = \frac{S+2}{N+4}$$

where p is the proportion, S is the number of times the color is mentioned, and N is the number of subjects (26 in the present research). Where 2 and 4 are experimentally determined constants, as described by Agressti and Coull [2].

The confidence interval was determined by:

$$p - \phi\sqrt{\frac{p\,(1-p)}{N+4}} \quad to \quad p + \phi\sqrt{\frac{p\,(1-p)}{N+4}}$$

where ϕ is 2.58 or 1.96 (in literature frequently rounded to 2.5 and 2 respectively) for the critical values from the Gaussian distribution for respectively 99% and 95%. The (relative) frequencies as well as the confidence intervals (both 99% and 95%) for all colors mentioned, are given in Table 3.2.

There were some observations of the experimenter of possible factors of influence on the data provided by the question of mentioning 10 colors:

- Most subjects were directly able to write down 7, 8, or 9 color names, but experienced it as difficult to mention the last.
- A considerable number of participants asked whether black, gray, and white were colors during their task of writing down 10 color names. This was confirmed by the researcher who conducted the experiment.
- Another group of subjects indicated after they had written down the color names that their opinion was that black, gray, and white are no colors. With that as opinion they had chosen to not write down black, gray, and white. This explains for a large part the less frequently mentioned colors, most written down last.

Table 3.2: Frequency and confidence-intervals of color names mentioned.

Color name	Frequency (in %)	min.-max. p at 99% (in %)	min.-max. p at 95% (in %)
red	26 (100.0%)	81.6% - 100.0%	84.4% - 100.0%
green	26 (100.0%)	81.6% - 100.0%	84.4% - 100.0%
yellow	26 (100.0%)	81.6% - 100.0%	84.4% - 100.0%
blue	26 (100.0%)	81.6% - 100.0%	84.4% - 100.0%
purple	24 (92.3%)	70.6% - 100.0%	74.5% - 98.8%
orange	22 (84.6%)	61.2% - 98.8%	65.7% - 94.3%
black	20 (76.9%)	52.5% - 94.1%	57.5% - 89.2%
white	20 (76.9%)	52.5% - 94.1%	57.5% - 89.2%
brown	20 (76.9%)	52.5% - 94.1%	57.5% - 89.2%
gray	15 (57.7%)	33.4% - 80.0%	38.9% - 74.4%
pink	11 (42.3%)	20.0% - 66.6%	25.6% - 61.1%
violet	06 (23.1%)	5.9% - 47.5%	10.8% - 42.5%
beige	04 (15.4%)	1.2% - 38.8%	5.7% - 34.3%
ocher	03 (11.5%)	0.9% - 34.2%	3.3% - 30.0%
turquoise	02 (7.7%)	2.7% - 29.3%	1.1% - 25.5%
magenta	02 (7.7%)	2.7% - 29.3%	1.1% - 25.5%
indigo	02 (7.7%)	2.7% - 29.3%	1.1% - 25.5%
cyan	02 (7.7%)	2.7% - 29.3%	1.1% - 25.5%
silver	01 (3.8%)	4.1% - 24.1%	0.7% - 20.7%
gold	01 (3.8%)	4.1% - 24.1%	0.7% - 20.7%
bordeaux-red	01 (3.8%)	4.1% - 24.1%	0.7% - 20.7%

As presented in Table 3.2, every subject named red, green, blue, and yellow. With 11 occurrences, pink was the least mentioned *focal color*. Nevertheless, pink was mentioned almost twice as much as the most frequently mentioned *non-focal color*: violet (6). The other *non-focal colors* were mentioned even less. In addition, the three observations mentioned above only confirm the existence of the *focal colors* in human memory.

3.3.2 The color discrimination and color memory experiment separate

The main result of both experiments is a table of markers for a CLUT. The full table of CLUT markers can be found in Appendix A. The table distinguishes the discrimination and memory experiment.

We have analyzed the color discrimination experiment on each of the three dimensions: R, G, and B. In each experiment, four blocks were present with the same randomized stimuli (see Section 3.2.4). The categorization of the same stimuli differed between the blocks ($p < .001$). This held for all 11 color categories. The same was done for the color memory experiment. Again block appeared a strong factor of influence ($p < .001$). Again this held for all 11 color categories.

Table 3.3: Differences between the color memory and the color discrimination experiment, per RGB color axis, per color category.

Color axis	color category	Strength and significance
R	blue	$F(1,25) = 3.48, p < .075$
	brown	$F(1,25) = 3.74, p < .065$
	purple	$F(1,25) = 6.49, p < .017$
	red	$F(1,25) = 20.50, p < .001$
G	black	$F(1,25) = 35.27, p < .001$
	blue	$F(1,25) = 35.46, p < .001$
	brown	$F(1,25) = 33.52, p < .001$
	green	$F(1,25) = 21.79, p < .001$
	orange	$F(1,25) = 30.12, p < .001$
	purple	$F(1,25) = 15.91, p < .001$
	red	$F(1,25) = 12.58, p < .002$
	white	$F(1,25) = 22.26, p < .001$
B	black	$F(1,25) = 12.89, p < .001$
	blue	$F(1,25) = 7.67, p < .010$
	brown	$F(1,25) = 8.67, p < .007$
	orange	$F(1,25) = 4.02, p < .056$
	pink	$F(1,25) = 9.82, p < .004$
	white	$F(1,25) = 7.19, p < .013$
	yellow	$F(1,25) = 7,67, p < .010$

3.3.3 The color discrimination and the color memory experiment together

The analysis of the experiments, conducted separately on each of the three dimensions: R, G, and B, showed a strong difference between the experiments on each of the three dimensions ($R : F(11,15) = 2.96, p < .027; G : F(11,15) = 7.843, p < .001; B : F(11,15) = 3.11, p < .022$). The results of a more detailed analysis for each color category separate on the R, G, and B dimensions of the RGB color space are provided in Table 3.3. The color categories that are not mentioned for each of the dimensions are not influenced by the difference in buttons between both experiments.

However, it is much more interesting to consider the colors independent of their (R, G, and B) dimensions. In both experiments (the overlap), 62 of the same web-safe colors were categorized as blue, 69 were categorized as green, and 49 were categorized as purple. Especially, for the color category purple, a clear difference between both experiments was present. The remaining colors were categorized to one of the other 9 color categories. The overlap between both experiments for these categories was much smaller (average: 12.89; range: 4-20). The differences were large (average: 6.78; range: 1-19).

3.4 Discussion

The questionnaire showed that the 11 color categories exist. This validated not only the choice of the 11 buttons used for the categorization of stimuli in the experiment, but, more importantly, it validated the idea to describe color space, using these color categories. When people use color categories when thinking, speaking, and remembering colors (see Chapter 2), why not use them for describing the color space and use this description for CBIR? Since the existence of color categories turned out to be valid we used them for two experiments on color categorization: specified for triggering processes related to respectively color discrimination and color memory.

Conform the hypothesis, no consistent color categorization was found *over* the experiments. This, despite the fact that the same stimuli were presented in the same blocks with the same button order, for each of the experiments; see the CLUT in Appendix A. So, this leaves as conclusion that the cognitive processes of discrimination and memory influence color categorization strongly.

The *CLUT*-markers were derived from the experimental results. They enable color matching using a human-based color space segmentation. Such an approach could enhance the color matching process significantly. Results based on such a color space description would be more intuitive for users. This would yield for the user more satisfying results than when using non-intuitive color matching functions founded on an arbitrary quantization of color space.

Furthermore, the strong effect of the stimulus order on their perception was remarkable, as indicated by the very strong influence of the factor block on the color categorization. This again indicates the strong influence of color memory on color perception. However, this did not explain that the *CLUT* markers define fuzzy boundaries between the color categories. This is due to a wide range of variables influencing color perception: memory, illumination, object identity, culture, emotion, and language, see Chapter 2.

So, points in RGB color space were assigned to human color categorizes, founded on two different cognitive processes: color discrimination and color memory. These categorized points can be considered as markers for an on human perception based division of color space. Hence, these markers provide the means for a color space segmentation and, subsequently, quantization that is based on human cognition.

In the next chapter, the concept of Weighted Distance Mapping (WDM) is introduced. This technique was used to segment the complete color space, based on the CLUT markers. In Chapter 5, the WDM is explained, which was applied on the CLUT markers, as presented in the present chapter. The latter resulted in the first complete description of color space based on human color categories. Moreover, this description can be utilized for CBIR matching purposes as will be shown later in this thesis.

4

Multi class distance mapping

Abstract

A new method is introduced for describing, visualizing, and inspecting data spaces. It is based on an adapted version of the Fast Exact Euclidean Distance (FEED) transform. It computes a description of the complete data space based on partial data. Combined with a metric, a true Weighted Distance Map (WDM) can be computed, which can define a confidence space. Subsequently, distances between data points can be determined. Using edge detection, borders (or boundaries) between categories (or clusters) of data can be found. Hence, Voronoi diagrams can be created. Moreover, the two-dimensional visualization of such WDMs provides excellent means for data inspection. Several examples illustrate the use of WDMs as well as their efficiency. So, a new, fast, and exact data analysis method has been developed that yields the means for a rich and intuitive method of data inspection.

This chapter is almost identical to:

Broek, E. L. van den, Schouten, Th. E., Kisters, P. M. F., and Kuppens H. (2005). Weighted Distance Mapping (WDM). In N. Canagarajah, A. Chalmers, F. Deravi, S. Gibson, P. Hobson, M. Mirmehdi, and S. Marshall (Eds.), *Proceedings of The IEE International Conference on Visual Information Engineering (VIE2005)*, p. 157-164. April 4-6, Glasgow - United Kingdom. Wrightsons - Earls Barton, Northants, Great Britain.

4.1 Introduction

With the increasing amounts of information in the current society, the need for data visualization and interpolations in data spaces becomes more and more important. In the last decades, both automated and manual procedures have been developed for these purposes. These procedures not seldomly rely on clustering techniques (introduced by [298]), used in a wide range of disciplines. The clusters obtained provide a way to describe the structure present in data, based on a certain feature representation. However, they rely on the availability of data that appropriate cover the corresponding data space.

In practice, often only partial data is available. Nevertheless, a description of the complete data space can be required. In such a case, approximations have to be made concerning those parts of the data space that lack data. This chapter presents a promising new approach: Weighted Distance Mapping (WDM), which provides the means to describe and inspect $\mathbb{Z}^2$ complete data spaces, based on any arbitrary number of data points.

As will be shown in this chapter, WDMs can be utilized for four purposes: (1) describe the complete data space based on a limited number of data points that fill only a small part of the complete data space, (2) rapid visualization of data spaces, (3) determine distances between data, using any metric, and (4) extraction of edges (or boundaries) between categories. These four features can be useful in a wide range of applications; e.g., robot navigation [143, 253] and segmentation of a data space based on valuable but limited experimental data (e.g., color assigned to color categories), as described in Section 4.7 and used by [30, 225].

Before WDM is introduced, morphological processing and the (Euclidean) distance transform are briefly described. In addition, the Voronoi diagram is briefly discussed, as a primitive distance map. Next, an adapted version of the algorithm that provides Fast Exact Euclidean Distance (FEED) transformations [254] is applied to obtain the WDMs.

In Section 4.5, FEED is compared with the city-block distance, as a baseline, and with Shih and Wu's [259] 2-scan method, as a state-of-the-art fast Euclidean distance (ED) transform. Next, FEED is applied on the confidence-based categorization of a color space, based on experimental data. This chapter ends with conclusions and a brief exposition of advantages and disadvantages of WDMs generated by FEED.

4.2 From morphological processing to distance transform

The operations dilation (also named dilatation) and erosion, illustrated in Figures 4.1 and 4.2, are fundamental to morphological processing of images. Many of the existing morphological algorithms are based on these two primitive operations [95].

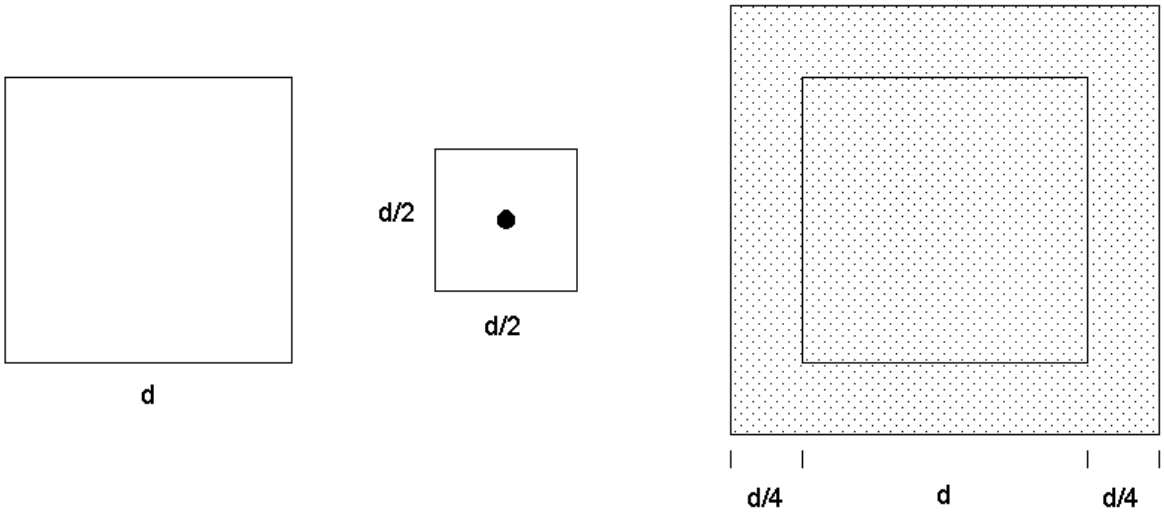


Figure 4.1: The process of dilation illustrated. The left figure is the original shape A. The square in the middle is the dilation marker B (dot is the center). The middle of the marker runs over the boundary of A. The result of dilation of A by B ($A \oplus B$) is given by the solid shape on the right, in which the inner square projects the original object A.

Given two sets A and B in $\mathbb{Z}^2$, the dilation of A by B, is defined as:

$$A \oplus B = \{x \,|\, (B)_x \cap A \neq \emptyset\}, \tag{4.1}$$

where $(B)_x$ denotes the translation of B by $x = (x_1, x_2)$ defined as:

$$(B)_x = \{c \,|\, c = b \,+\, x, \text{for some } b \in B\} \tag{4.2}$$

Thus, $A \,\oplus\, B$ expands A if the origin is contained in B, as is usually the case.

The erosion of A by B, denoted $A \ominus B$, is the set of all x such that B translated by x, is completely contained in A, defined as

$$A \ominus B = \{x \,|\, (B)_x \subseteq A\} \tag{4.3}$$

Thus, $A \,\ominus\, B$ decreases A.

Based on these two morphological operations the *4-n* and the *8-n* dilation algorithms were developed by Rosenfeld and Pfaltz [233] for region growing purposes. These region growing algorithms are based on two distance measures: the city-block distance and the chessboard distance. The set of pixels contained in the dilated shape, for respectively *4-n* and *8-n* growth for an isolated pixel at the origin, are defined as:

$$C_4(n) \,=\, \{\,(x,y) \,\in\, \mathbb{Z}^2 : \,|x| \,+\, |y| \,\leq\, n\,\}, \tag{4.4}$$

$$C_8(n) \,=\, \{\,(x,y) \,\in\, \mathbb{Z}^2 : \,|x| \,\leq\, n,\, |y| \,\leq\, n\,\}, \tag{4.5}$$

where n is the number of iterations.

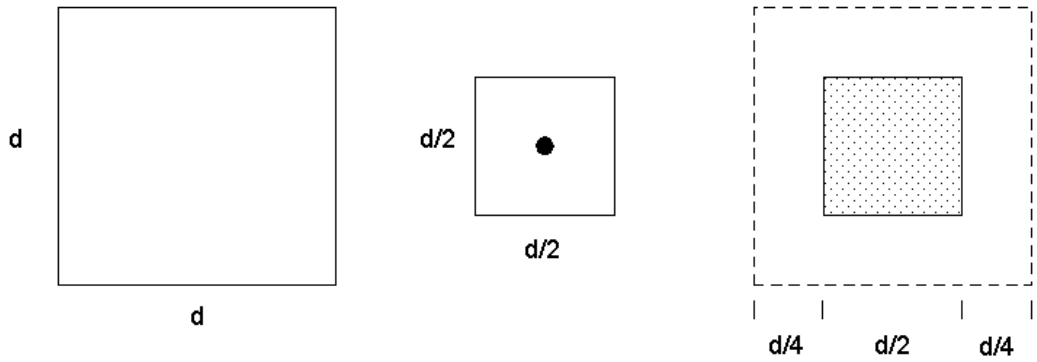


Figure 4.2: The process of erosion illustrated. The left figure is the original shape A. The square in the middle is the erosion marker B (dot is the center). The middle of the marker runs over the boundary of A. The result of erosion of A by B ($A \ominus B$) is given by the solid shape on the right, in which the outer (dotted) square projects the original object A.

To obtain a better approximation for the ED, Rosenfeld and Pfaltz [233] recommended the alternate use of the city-block and chessboard motions, which defines the octagonal distance. The octagonal distance provides a better approximation of the ED than the other two distances.

Thirty years later Coiras et al. [58] introduced hexadecagonal region growing, again a combination of *4-n* and *8-n* growth (see Figure 4.3). The latter uses the identification of vertex pixels for vertex growth inhibition. This resulted in an approximation of ideal circular region growing up to 97.3% and, hence, outperformed six other growth models.

4.3 Euclidean Distance transformation (EDT)

Region growing algorithms can be applied to obtain distance transformations. A distance transformation [233] creates an image in which the value of each pixel is its distance to the set of object pixels O in the original image:

$$D(p) = \min\{\text{dist}(p, q), q \in O\} \tag{4.6}$$

The Euclidean distance transform (EDT) has been extensively used in computer vision and pattern recognition, either by itself or as an important intermediate or ancillary method in applications ranging from trajectory planning [324] to neuromorphometry [62]. Examples of methods possibly involving the EDT are: (i) skeletonization [144]; (ii) Voronoi tessellations [99]; (iii) Bouligand-Minkowsky fractal dimension [61], (iv) quadratic structuring functions [19, 20], (v) Watershed algorithms [183], (vi) wave propagation [78], and (vii) robot navigation [143, 253]. For example, in the next chapter the EDT is applied to gener-

```
for k:=1 to R
    for every pixel p in the boundary
        if NOT [(p is a vertex) AND (k modulo 5=0)
            AND (k modulo 45!=0)]
            if [(k modulo 2=0) AND (k modulo 12!=0)
                AND (k modulo 410!=0)]
                grow p as 8-n
            otherwise
                grow p as 4-n
```

Figure 4.3: Algorithm for hexadecagonal growth (source: [58]). R denotes the number of iterations.

ate Voronoi diagrams. In addition, recently Schouten, Kuppens, and Van den Broek [253] applied the EDT for robot navigation purposes.

Several methods for calculation of the EDT have been described in the literature [20, 86, 251, 253, 289], both for sequential and parallel machines. However, most of these methods do not produce exact distances, but only approximations [66]. Borgefors [21] proposed a chamfer distance transformation using two raster scans on the image, which produces a coarse approximation of the exact EDT. To get a result that is exact on most points but can produce small errors on some points, Danielsson [69] used four raster scans.

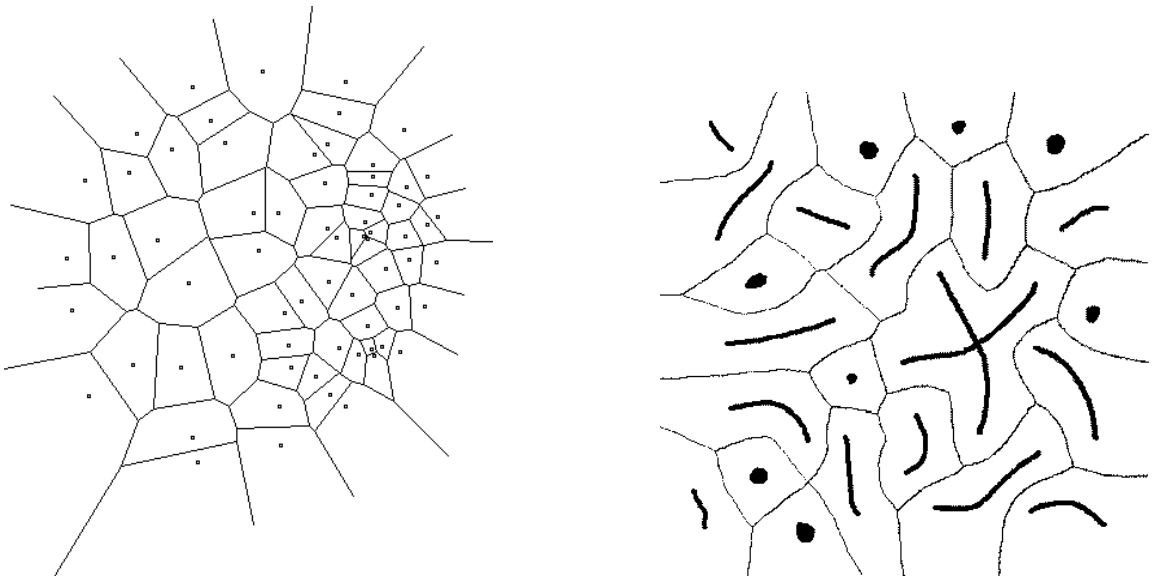
In order to obtain an exact EDT, two step methods were proposed. Two of the most important ones are:

- Cuisenaire and Macq [66] first calculated an approximate EDT, using ordered propagation by bucket sorting. It produces a result similar to Danielsson's. Second, this approximation is improved by using neighborhoods of increasing size.
- Shih and Liu [258] started with four scans on the image, producing a result similar to Danielsson's. A look-up table is then constructed containing all possible locations where no exact result was produced. Because during the scans the location of the closest object pixel is stored for each image pixel, the look-up table can be used to correct the errors. Shih and Liu claim that the number of error locations is small.

4.3.1 Voronoi diagrams

Exact EDTs can be applied to obtain distance maps such as the Voronoi diagram (see Figure 4.4)[1]. The Voronoi diagram $V(P)$ is a network representing a plane subdivided by the influence regions of the set of points $P = \{p_1, p_2, ..., p_n\}$. It is constructed by a set of Voronoi

[1]The Voronoi web page: http://www.voronoi.com



(a) (b)

Figure 4.4: Voronoi diagrams of (a) a set of points and (b) a set of arbitrary shapes as determined by way of region growing.

regions $V(p_i)$ which is, for any i, defined by

$$V(p_i) = \{x \in \mathbb{Z}^2 : |x - p_i| \leq |x - p_j|, \text{ for all } j\} \tag{4.7}$$

Voronoi diagram generation of a space with arbitrary shapes (see Figure 4.4) is hard from the analytical point of view [7, 12], but is easily solved by applying a growth algorithm.

4.4 Fast Exact Euclidean Distance (FEED)

In contrast with the existing approaches such as those of Shih and Liu [258] and Cuisenaire and Macq [66], we have implemented the EDT starting directly from the definition in Equation 4.6. Or rather its inverse: each object pixel q, in the set of object pixels (O), *feeds* its ED to all non-object pixels p. The naive algorithm then becomes:

initialize $D(p) =$ if $(p \in O)$ then 0, else ∞
  foreach $q \in O$
    foreach $p \notin O$
      update : $D(p) = \min(D(p), \mathrm{ED}(q, p))$

However, this algorithm is extremely time consuming, but can be speeded up by:

- restricting the number of object pixels q that have to be considered
- pre-computation of $\mathrm{ED}(q, p)$
- restricting the number of background pixels p that have to be updated for each considered object pixel q

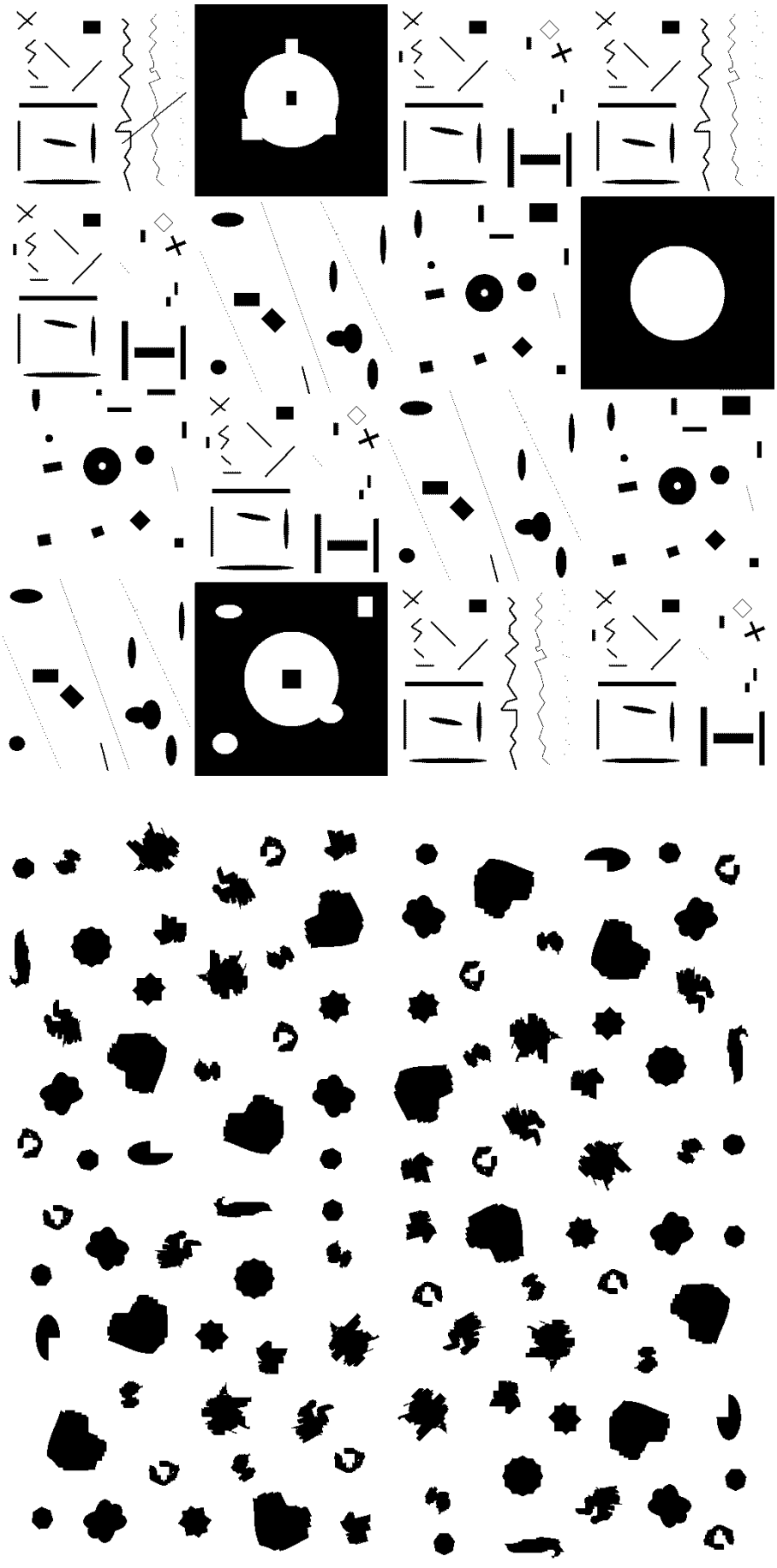


Figure 4.5: Two examples of the test images used for the comparison of the city-block (or Chamfer 1,1) transform, Shih and Wu's 2-scan method (EDT-2), and the Fast Exact Euclidean Distance (FEED) transform.

This resulted in an exact but computationally less expensive algorithm for EDT: the Fast Exact Euclidean Distance (FEED) transformation. It was recently introduced by Schouten and Van den Broek [254] (available as Appendix C). For both algorithmic and implementation details we refer to this paper. For an improved version, named timed FEED (tFEED), and an extension to fast handling of video sequences, we refer to [253]. Recently, FEED was redefined such that it can be applied independent of the number of dimensions needed/present [251] and introduces the three dimensional version of FEED (3D-FEED).

In its naive implementation, FEED proved already to be up to $3\times$ faster than the algorithm of Shih and Liu [258]. Providing that a maximum distance in the image is known a priori, it is even up to $4.5\times$ faster.

To be able to utilize FEED for the creation of WDMs, we have applied a few small modifications to the implementation, compared to the algorithm as introduced in Schouten and Van den Broek [254]. FEED was adapted in such a way that it became possible to define a metric on which the WDM was based. The result of the application of various metrics is illustrated in Figures 4.6, 4.7, and 4.8.

4.5 Benchmarking FEED

Shih and Wu describe in their paper "Fast Euclidean distance transformation in two scans using a 3×3 neighborhood" [259] that they introduce an exact EDT. They propose their algorithm as the, to be preferred, alternative for the fast EDT as proposed by Cuisenaire and Macq [66]. Shih and Wu's algorithm is the most recent attempt to obtain fast EDTs. Therefore, this algorithm would be the ultimate test for our FEED algorithm.

We have implemented Shih and Wu's algorithm (EDT-2) exactly as they described and tested it on a set of eight images on both processing time and errors in the EDs obtained, see Figure 4.5 for two example text-images. As a baseline, the city-block (or Chamfer 1,1) distance was also taken into account.

In Table 4.1, the timing results can be found for the city-block measure, for Shih and Wu's two scans (EDT-2), and for FEED. As was expected, with a rough estimation of the ED, the city block distance outperformed the other two algorithms by far (see Table 4.1). More surprising was that FEED was more than twice as fast as EDT-2. However, the aim of this research was to utilize exact EDT. Hence, next to the timing results, the percentage of errors made in obtaining the ED is of interest to us. The city-block transform resulted for the set of eight images in an error-level of less than 5%; see Table 4.2. Shih and Wu's claimed that their two scan algorithm (EDT-2) provided exact EDs. In 99% of the cases their claim appeared justified. However, errors occur in their algorithm, which are reported in Table 4.2. So, FEED appeared to be the only algorithm that provided the truly exact ED for all instances.

Table 4.1: Timing results for a set of eight images on the city-block (or Chamfer 1,1) transform, Shih and Wu's 2-scan method (EDT-2), and for the Fast Exact Euclidean Distance (FEED) transform.

Images	Algorithms		
	City-block	EDT-2	FEED
standard	8.75 s	38.91 s	17.14 s
rotated	8.77 s	38.86 s	18.02 s
larger obj.	8.64 s	37.94 s	19.94 s

Table 4.2: Errors of the city-block (or Chamfer 1,1) transform and of Shih and Wu's two scan algorithm (EDT-2). Note that no errors of FEED were observed since FEED provides truly exact EDs.

Images	Algorithms	
	City-block	EDT-2
standard	2.39%	0.16%
rotated	4.66%	0.21%
larger obj.	4.14%	0.51%

4.6 Weighted distance mapping (WDM)

This section describes the WDM method. Distance maps in general are, for example, used for skeletonization purposes [157] or for the determination of pixel clusters. Given a metric, WDM provides a distance map representing a distance function, which assigns a weight to all of the points in space. Such a weight can be a confidence measure, for example, determined by $\frac{1}{\mathrm{ED}}$, $\sqrt{\mathrm{ED}}$, or $\log(\mathrm{ED})$.

So, using distance functions, distance maps can be created [58]. This is done by growth models based on these distance functions. These distance maps give an excellent overview of the background-pixels that are close to a certain object pixel: A distance map divides the space in a set of regions, where every region is the set of points closer to a certain element than to the others.

4.6.1 Preprocessing

In order to reduce processing time or to enhance the final WDM, preprocessing algorithms of choice can be applied. For instance, noise reduction and pre-clustering algorithms.

In the first case, a range can be within which is scanned for other points. If no points with the same label are found within this range, this point is rejected as input for WDM.

In the second case, when data points having the same label, are within a range (as was provided), and no other points with another label lay between, then the data points can

be connected. When this is done for all data points with the same label, a fully connected graph is generated for this cluster of points. Next, the same label can be assigned to all points within this fully connected graph. Hence, instead of labeled data points, labeled objects serve as input for the WDM.

4.6.2 Binary data

Let us first consider a set of labeled binary data points; i.e., each point in space either is or is not an object point. An isolated pixel with value *0* at the origin is grown using FEED, up to a chosen radius. Each grown pixel then receives a value according to its distance to the origin. As default the ED is used, but any metric could be used. The resulting image defines a mask B.

The output image is initialized with the input image, assuming *0* for an object pixel and a maximum value for a background pixel. Then a single scan over the input image A is made. On each pixel of A with value *0* (an object pixel) the mask is placed. For each so covered pixel, the output value is updated as the minimum of the current value and the value given by the mask.

The resulting output image contains then for each background pixel its minimum distance to the set of object pixels according to the metric of choice. In the case of binary data, the WDM can be stored in one matrix. In Figure 4.6, some results of the WDM, using different metrics, are shown using the same input image as Coiras et al. [58].

4.6.3 Multi class data

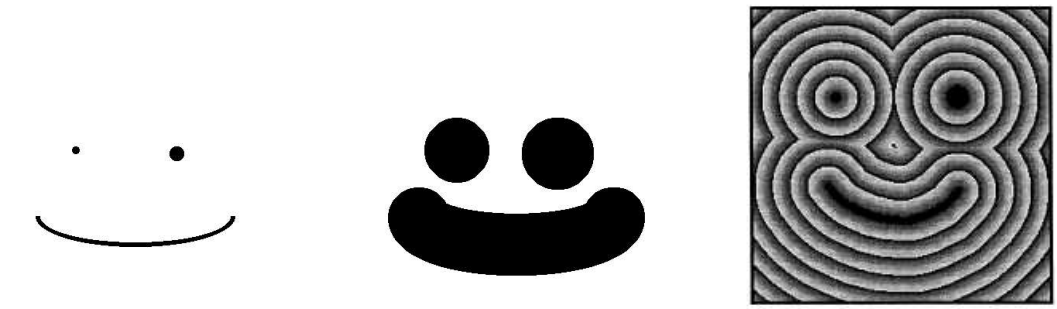
Now, let us consider the case that multiple labeled classes of data points are present and, subsequently, WDM is applied for data space segmentation. In such a case, the class of the input pixel that provides the minimum distance can be placed in a second output matrix.

The minimum distance value then indicates the amount of certainty (or weight) that the pixel belongs to the class. This can be visualized by different color ranges, for each class. In addition, a hill climbing algorithm can be applied, to extract edges from the distance image and so generate a Voronoi diagram (see Section 4.3.1).

To determine the class to which the ED is assigned, the update step of FEED was changed to:

$$\begin{aligned} \text{update}: \ & \text{if } (\text{ED}(q,p) \ < \ D(p)) \\ & \text{then } (D(p) \ = \ \text{ED}(q,p);\ C(p) \ = \ C(q)\ ) \end{aligned}$$

where C is a class matrix, in which all data is assigned to one of the classes. Figure 4.7 illustrates the latter algorithm. It presents a set of six arbitrary shapes, their ED maps, and



(a) (b) (c)

(d) (e) (f)

Figure 4.6: (a) is the original image. (b) is the same image after dilation by hexadecagonal region growing. (c) is a distance map as presented by Coiras et al. [58]. (d), (e), and (f) are weighted distance maps (WDM). (d) provides the extremes (i.e., the original pixels and the boundary of the dilated image). (e) presents a discrete distance map. (f) presents an gradual decrease in weight with the increase of distance from the original pixels.

the classification as provided. The combination of the ED maps and the classification are the input for a true WDM. The resulting WDM can serve four purposes:

1. It provides a confidence space. The complete data space is described by providing certainties to unknown regions in the data space; e.g., $\frac{1}{\mathrm{ED}}$, $\sqrt{\mathrm{ED}}$, $\log(\mathrm{ED})$. This results in fuzzy boundaries (see Figures 4.6e-f, 4.7b,d, and 4.8b).
2. Determination of the edges between categories (see Figure 4.7c,d). In addition, Voronoi diagrams can be generated (see Figure 4.4a-b and cf. Figures 4.8a and 4.8c).
3. Distances between data can be determined, using any metric (see Figures 4.6c-f and 4.8d).
4. Visualization of the categorized (fuzzy) data space, as illustrated in Figures 4.6e-f, 4.7b,d, and 4.8b.

Figure 4.7: From left to right and from top to bottom: the original image, the basic ED map, the fully segmented and labeled space, and the labeled Weighted Distance Map (WDM), in which the original objects are projected and where pixel intensities denotes the weight.

4.7 An application: segmentation of color space

WDM as described in the previous section has been validated on various data sets (see for example Figure 4.6 and 4.7). We will illustrate its use for the categorization of color in the 11 color categories (or focal colors) [30, 75, 219, 225]. This was one of the data sets on which the mapping was validated. For more information on the topic of the 11 color categories see, for example, the World Color Survey [140].

The clustered data is derived from two experiments that confirmed the existence of the 11 color categories [28], as described in the previous chapter. The Color LookUp Table markers that resulted from the experiments were RGB coordinates. These were converted to HSI-coordinates [89]. Let us consider the Hue and Saturation axes of the HSI-color space, using a slice of the HSI cylinder. In this slice, five color categories (i.e., brown, red, purple, blue, and green) are projected. However, only four clusters are present. This is due to the overlap between the color categories red and brown.

A bitmap image was generated, containing white background pixels and labeled pixels representing each of the data points. For each category, the data points belonging to the same cluster, were fully connected by using a line generator, as shown in Figure 4.8a. Next, WDM was applied on the image; see Figure 4.8b. This resulted in two matrices. One of them consists of the weights determined; in the other matrix the class each point is assigned to, is stored. Their combination provides the ED map.

Last, a hill climbing algorithm extracted edges from the ED map, as shown in Figure 4.8c. On the one hand, this resulted in fuzzy color categories (providing certainties). On the other hand, the extracted edges define a Voronoi diagram.

Since a few years the interest in color in the field of image processing exploded. An ED map as presented, based on experimental data, provides an excellent way for describing the color space. Next, the perceptual characteristics of the color categories could be exploited, providing a confidence distribution and, subsequently, a metric for each of the color categories separate. Such a set of features can be utilized and can in combination with a ED map, provide a true WDM.

(a) (b) (c)

Figure 4.8: (a) The original image in which all data points (of four color categories) assigned to the same color category are connected with each other, using a line connector. (b) The basic ED map of (a), in which the intensity of the pixels resembles the weight. (c) The boundaries between the four classes, derived from the ED map as presented in (b). A hill climbing algorithm was used to extract these boundaries. Note that (c) is the Voronoi diagram of (a).

4.8 Discussion

This chapter started with a brief overview of morphological operations and distance transforms. Next, the algorithm which generates a Fast Exact Euclidean Distance (FEED) transform, is introduced. It is compared with the city-block measure (as baseline) and with the two scan algorithm of Shih and Wu [259], which can be considered as a state-of-the-art algorithm on fast exact ED transforms. FEED proved to be computationally twice as cheap as Shih and Wu's algorithm. Moreover, in contrast with the algorithm of Shih and Wu, FEED provides in all cases exact EDs. FEED is applied to generate Weighted Distance Maps (WDM), providing a metric. Its use is illustrated by the segmentation of color space, based on a limited set of experimentally gathered data points. WDMs, as proposed, provide complete descriptions of data spaces, based on a limited set of classified data. Moreover, they can be used to obtain Voronoi diagrams.

With the launch of a parallel implementation of FEED (tFEED) [253], the generation of WDMs can probably be boosted in the near future. So, WDMs four main advantages can be exploited even more: (1) A complete data space can be described, based on a limited set of data points, (2) Data spaces can be visualized rapidly, providing the possibility to explore the data space gaining more understanding, (3) Distances between data can be determined, using any metric, and (4) Edges between categories can be determined. These features make WDM an intuitive, flexible, and powerful tool for data visualization and interpolations in data spaces, describing data spaces, and for either fuzzy or discrete data space segmentation.

In Section 4.7 of this chapter, a WDM was generated based on Color LookUp Table (CLUT) markers projected in a slice of the HSI color space. In addition, a Voronoi diagram was extracted from this WDM, providing the edges between the color categories in the slice. In the next chapter, we will describe how WDM was applied on the complete set of CLUT markers, as presented in the previous chapter. These will be pre-processed, fed to WDM, and post-processed, in a manner such as introduced in this chapter.

5

Efficient Human-centered Color Space Segmentation

Abstract

A unique color space segmentation method is introduced. It is founded on features of human cognition, where 11 color categories are used in processing color. In two experiments, human subjects were asked to categorize color stimuli into these 11 color categories, which resulted in markers for a Color LookUp Table (CLUT). These CLUT markers are projected on two 2D projections of the HSI color space. By applying the newly developed Fast Exact Euclidean Distance (FEED) transform on the projections, a complete and efficient segmentation of color space is achieved. Thus, a human-based color space segmentation is generated, which is invariant for intensity changes. Moreover, the efficiency of the procedure facilitates the generation of adaptable, application-centered, color quantization schemes. It is shown to work well for color analysis, texture analysis, and for Color-Based Image Retrieval purposes and is, for example, suitable for applications in the medical and cultural domain.

This chapter is an adapted version of:

Broek, E. L. van den, Schouten, Th. E., and Kisters, P. M. F. (2005). Efficient human-centered color space segmentation. *[submitted]*

5.1 Introduction

Digital imaging technology is more and more embedded in a broad domain. Consequently, digital image collections are booming, which creates the need for efficient data-mining in such collections. An adequate model of human visual perception would facilitate data-mining. Our approach, hereby, is to utilize human cognitive and perceptual characteristics.

In this chapter, we will focus on a generic image processing technique: A color quantization scheme based on human perception. This unique color space segmentation is both relevant and suitable for the development and study of content-based image retrieval (CBIR) in the context of rapidly growing digital collections in libraries, museums, and historical archives as well as in medical image collections [31].

We argue that in general color should be analyzed from the perspective of human color categories. Both to relate to the way people think, speak, and remember color and to reduce the data from 16 million or more colors to a limited number of color categories: black, white, red, green, yellow, blue, brown, purple, pink, orange, and gray [75]. People use these categories in general when thinking, speaking, and remembering colors. Research from diverse fields of science emphasize their importance in human color perception. The use of this knowledge can possibly provide a solution for problems concerning the accessibility and the availability of knowledge, where color analysis is applied in data-mining. In addition, such a human-centered approach can tackle the computational burden of traditional (real-time) color analysis [85, 136].

The 11 color categories are applicable for a broad range of CBIR domains, where in specific (i.e., specialized) domains, other sets of colors might be more appropriate. In this chapter, we regard the 11 color categories as they are used in daily life (see also Chapter 2). These color categories are constructed and handled by methods that are presented in this chapter. However, in the same way, it is possible to incorporate another set of colors, which is user, task, or application specific.

This chapter presents a line of research starting with the psychophysical experiments, described in Chapter 3. This provided us with markers for a Color LookUp Table (CLUT) in the RGB color space. The boundaries between the color categories in the RGB space are expected to be too complex to be determined, using the limited number of CLUT markers. Therefore, we describe in Section 5.2 how the RGB space is transformed into two 2D projections of the HSI color space in which the boundaries are less complex. In Sections 5.3 and 5.4, we describe how the CLUT markers are used to find the boundaries between the color categories in the 2D projections and how this is used to segment the complete color space. For this, the Fast and Exact Euclidean Distance (FEED) transformation is used, which was briefly introduced in the previous chapter. In Section 5.5, the CLUT markers are compared with the segmented color space. Last, we draw some final conclusion in Section 5.6.

5.2 Preprocessing of experimental data

In general, color matching using a CLUT, based on the markers derived from the experimental results described in Chapter 3, could enhance the color matching process significantly and may yield more intuitive values for users [75]. In addition, such a coarse color space quantization of 11 color categories reduces the computational complexity of color analysis drastically, compared to existing matching algorithms of image retrieval engines that use color quantization schemes (cf. PicHunter [63]: HSV $4\times4\times4$ and QBIC [85]: RGB $16\times16\times16$). The coarse 11 color categories quantization makes it also invariant with respect to intensity changes. The experiments presented in Chapter 3, provided us with categorized markers for a CLUT. In this section, we explain the preprocessing scheme that transforms these markers in order to facilitate segmentation of the complete color space.

5.2.1 From the RGB CLUT markers to the HSI CLUT markers

The markers of the CLUT are RGB coordinates; however, the RGB color space is not perceptually intuitive (see also Chapter 2. Hence, the position and shape of the color categories within the RGB color space are complex. Moreover, there are too little CLUT markers (see also Appendix A) to directly determine the complex boundaries between the categories in the RGB space. Therefore, for the full color space categorization, the HSI color space is used, which is (i) perceptually intuitive, (ii) performs as good as or better than perceptual uniform color spaces such as CIE LUV [159], and (iii) the shape and position of the color categories are less complex functions of location and orientation, than with the RGB color space.

Let us now briefly discuss the HSI color space. The axes of the HSI space represent hue (i.e., basic color index), saturation (i.e., colorfulness), and intensity (i.e., amount of white present in the color). The shape of HSI color space can be displayed as a cylinder: intensity is the central rod, hue is the angle around that rod, and saturation is the distance perpendicular to that rod. The color categories' orientation is as follows: Around the intensity axis, the achromatic categories (i.e., black, gray, white) are located. The achromatic region has the shape of an hourglass and is described with small saturation values, the complete range of intensity, and the complete range of hue values. Around this achromatic region, the chromatic categories are located. Chromatic categories have high saturation values and occupy a part of both the total hue and the total intensity range.

The 216 web-safe colors are clearly distinct for human perception. As a consequence, in a perceptually intuitive color space as the HSI color space is, some distance is present between them. Moreover, the perceptually intuitive character of the HSI color space results in an orientation of adjacent colors such that the web-safe colors are arranged by color category.

The first phase in preprocessing is the conversion of the RGB CLUT markers to HSI

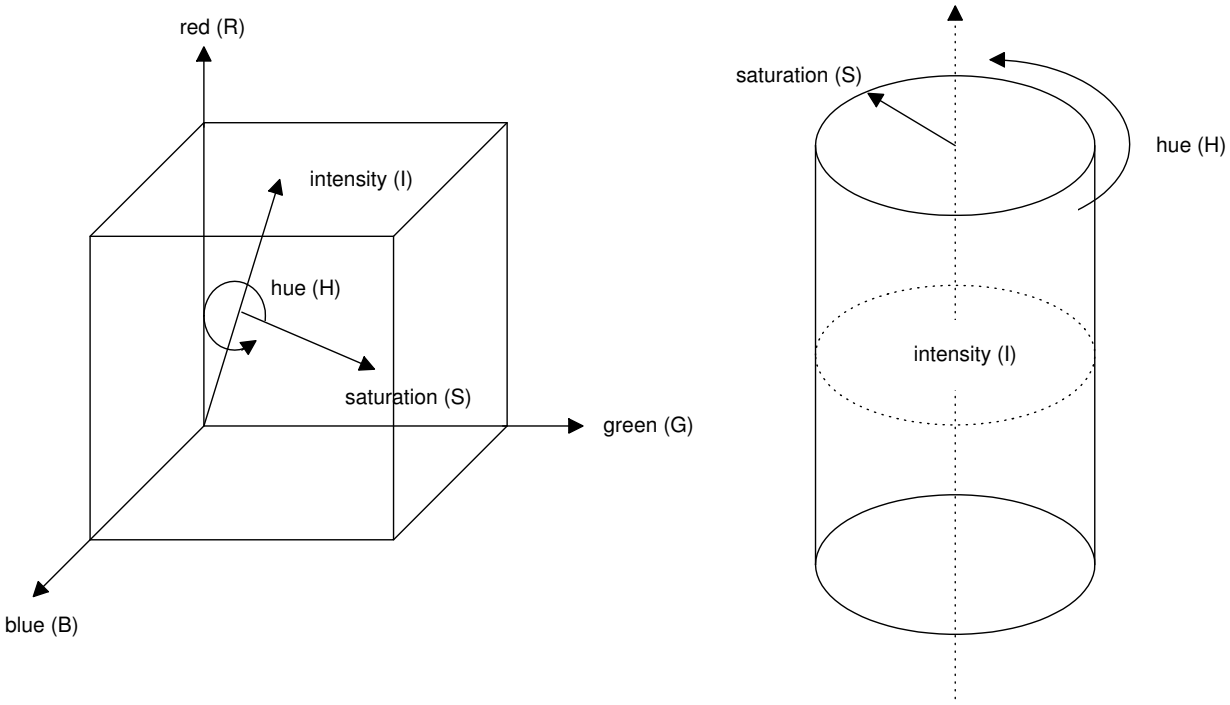


Figure 5.1: Left: The relation between the RGB and the HSI color space, from the perspective of the RGB color space. Right: The cylinder shaped representation of the HSI (hue, saturation, and intensity) color space, as used in this research.

CLUT markers. The conversions as defined by Gevers and Smeulders [89] are adopted. In Figure 5.1 the RGB and HSI color spaces are visualized. In the RGB color space, the HSI axes are denoted.

5.2.2 From 3D HSI color space to two 2D representations

Since the HSI color space is a 3D space, the boundaries between color categories consist of 3D functions. Unfortunately, the amount of HSI CLUT markers is too limited to determine the exact boundaries. Since some categories are expressed by only a few data CLUT markers in color space, 3D segmentation based on these markers would evolve in very weak estimations of the shape of color categories in color space.

However, the perceptually intuitive axes of the HSI color space do allow a reduction in the complexity of boundary functions without loosing essential features of the boundaries. The intuitive values that the axes represent, provided the means to separate chromatic and achromatic categories using two 2D projections. As a result, the extracted boundaries are functions in a 2D plane. Thereby, we use three assumptions:

1. The boundaries between achromatic categories and chromatic categories do not excessively change over the hue range.

2. The boundaries between chromatic categories do not excessively change over the saturation axis and can be approximated by a linear function toward the central rod of the color space; i.e., the intensity axis. The intuitive features of the HSI space provide strong arguments for the latter assumption: Consider a chromatic point of the outer boundaries of the HSI space (with maximum saturation). When the saturation value is lowered, the perceived color becomes 'decolorized' or pale. Nevertheless, in general the colors are perceived as belonging to the same color category.
3. The two boundaries between achromatic categories are each expressed with a single intensity value.

The next phase in preprocessing is the generation of the 2D planes and, subsequently, perform the segmentation in three steps: (i) separation of chromatic categories from achromatic categories, (ii) segmentation of the individual chromatic categories, and (iii) segmentation of the individual achromatic categories. So, the 11 color categories of the HSI CLUT were divided into two groups: the achromatic categories (i.e., black, gray, and white) and the chromatic categories (i.e., blue, yellow, green, purple, pink, red, brown, and orange). Note that each group is processed in a separate 2D plane; see Figures 5.3 and 5.4.

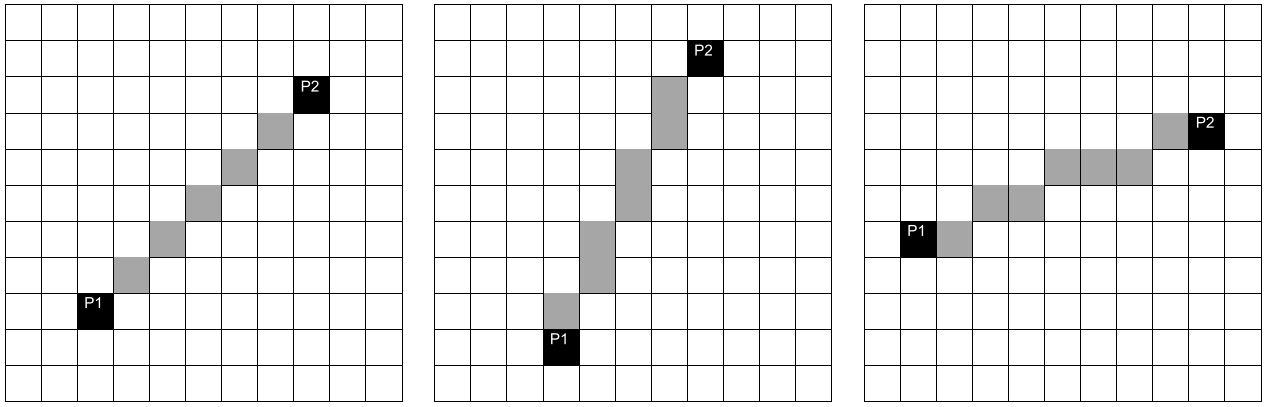
First, the achromatic categories were separated from the chromatic categories in a 2D plane leaving out the hue axis resulting in a saturation-intensity plane. In this projection, the achromatic categories are distinguished from the chromatic categories as a line and a cloud of data points (see Figures 5.3a and 5.3b). Note that, when leaving out the hue axis, the main color information is left out and thus, all individual chromatic categories resemble a single cloud of data points.

Second, the segmentation of the chromatic colors is done by leaving out the saturation axis: the hue-intensity plane. In this plane, the chromatic category data is projected. The result is a plane with non-overlapping clouds of categorized points, as illustrated in Figure 5.4a and 5.4b.

Third, the segmentation of the individual achromatic categories is performed. Since these categories do not represent any basic color information, the hue axis does not contain useful information for these categories. Thus, the segmentation of these individual achromatic color categories is done in a saturation-intensity plane (see Figure 5.3).

5.2.3 Labeling and connecting the HSI CLUT colors

Given these two 2D planes, boundary functions with a relatively low complexity are defined, resulting in a computationally cheap, complete color space segmentation. The HSI CLUT markers were plotted in the 2D planes discussed previously. In the next section, we describe how distance transforms were applied to segment these 2D planes. In order to facilitate



(a) (b) (c)

Figure 5.2: Connecting two data points $p1$ and $p2$ (with different slopes). (a) rise/run = 1, (b) rise/run = 2 and, (c) rise/run = 0.5.

this process, two final preprocessing steps were applied in the 2D planes: (i) for each color category, a fully connected graph is generated, using a line generator (cf. Figure 5.3a and Figure 5.3b as well as Figure 5.4a and Figure 5.4b) and (ii) these graphs were filled resulting in a convex hull to speed up the distance mapping (see Figures 5.3c and 5.4c).

For each color category, the fully connected graph was generated by connecting the CLUT markers, by means of a line connector algorithm. For each unique pair of CLUT markers points ($p1(x_{p1}, y_{p1})$ and $p2(x_{p2}, y_{p2})$), belonging to the same color category and situated in the matrix[x][y], the line connector algorithm calculates coordinates that represent the connection line between these points. These coordinates were calculated by using the rise/run quotient between $p1$ and $p2$. Next, the coordinates were rounded to integers in order to display them in the matrix (see Figure 5.2).

The clouds of data points in the 2D projections resemble ellipse-like shapes, which are mimicked by the boundaries of the connected graphs of the color categories. Hence, for each category we can assume that all points within the boundaries of the connected graphs belong to the color category to which all individual data points were assigned. Subsequently, the filled convex hulls are the initial estimation of the color categories within the HSI color space.

5.3 Using distance maps

The CLUT markers resulting from the experiments, discussed in Chapter 3, are preprocessed as described in Section 5.2. This resulted in a set of filled convex hulls in two 2D planes. Because the information that is available about human color categorization does not classify all possible points in color space, we applied distance mapping, where each point gets a

distance measure to the set of categorized points by humans. This assignment (i.e., the color quantization scheme) has to be both good and fast.

The speed of distance transforms is determined by the precision needed. Images are a priori an approximation of reality due to various forms of noise. It might be that introducing additional approximations in the image processing chain to make it faster, has no effect on the final quality of the application. But, how to prove that? Since assessing the validity of this assumption is a difficult and subjective process, our goal was to design a distance transform, which performs as accurate as possible, preferably exact.

Distance transforms can be applied to all kinds of data. In this chapter, we discuss the 11 color categories, which are generally applicable, as discussed in Chapter 2. However, the categories that are needed depend on the specific application; e.g., a catalog of paintings or a stained cell tissue database. There might be the need to adapt the color categories quickly to specifications based on a certain domain or task. Moreover, since the perception of individual users differs, systems are needed that employ user profiles [307], which would be in our case: a user specific color space segmentation. The latter is of great importance since users are in interaction with the systems, which use image analysis techniques, and judge their results. Therefore, we wanted a fast color space segmentation regarding computer and human resources.

The distance transform to be applied both needs to be fast enough and preferably exact. For this purpose, we applied the Fast Exact Euclidean Distance (FEED) transform, which was briefly introduced in the previous chapter. If preferred, a particular application can later plug in a faster and less accurate distance transform to determine its speed-quality curve and choose a setting.

5.4 Segmentation, post processing and utilization

Using FEED (see Chapter 4 and Appendix C), two distance maps were generated, one for each 2D plane of the HSI color space (see Section 5.2 and Figures 5.3d and 5.4d). A hill climbing algorithm is applied to determine the edges between the color categories (see Figure 5.3e and 5.4e). Next, by way of curve fitting techniques, the extracted edges were converted to Fourier functions that express the borders between the color categories. This approach is applied to both 2D planes: (i) the saturation-intensity plane, in order to separate the achromatic from chromatic categories (see Figure 5.3) and (ii) the hue-intensity plane to segment the individual chromatic categories (see Figure 5.4). Finally, segmentation of the achromatic colors is conducted in order to have a completely segmented color space. A drawback for the differentiation between the achromatic colors is the lack of achromatic CLUT markers. We take two intensity values that describe the boundaries between individual achromatic

categories in three sections of equal length.

With the latter step, the complete HSI space is segmented into the 11 color categories. One Fourier function expresses the boundary between the chromatic and achromatic region in a saturation-intensity plane (see Figure 5.3e). In total, 16 Fourier functions express the boundaries between the chromatic categories in the hue-intensity plane (see Figure 5.4e).

The boundaries are stored in an intensity matrix of size 765 by 17. With these $13,005$ coordinates, a computational cheap categorization of 256^3 color values into the 11 categories is done as follows: Given the intensity value of some HSI value that is being categorized, 17 $(16 + 1)$ border values are retrieved, which are used for the categorization process. One border value, representing a saturation value, is used for global chromatic/achromatic categorization (see Figure 5.3e). In case of a chromatic color, further processing of 16 numbered border values, is done to estimate the chromatic category (see Figure 5.4e). Each of these border values contain either a hue value or a dummy value when no border exists for that intensity range.

Please note, that with this implementation for some intensity value maximal six chromatic categories (boundaries of categories) are considered: first, the border for chromatic-achromatic categorization (Figure 5.3e); second, when going from left to right through Figure 5.4e: borders 3, 12, 6, 4, and 1. So, the color quantization scheme has a computational complexity that is lower-or-equal to a virtual HSI $6 \times 2 \times 1$ quantization scheme. In case of an achromatic color, the achromatic category is determined by means of the intensity value.

The presented implementation is a compact representation of the segmented HSI color space: it requires only a few computing steps to categorize a RGB value. From that representation, a fast categorization mechanism is easily established; by filling a 256^3 table with categorized RGB values, a fast, complete CLUT is available.

(a) (b) (c) (d)

Figure 5.3: The processing scheme of the separation of the chromatic from the achromatic color categories, in the saturation-intensity (SI) plane, using human color categorization data (see Chapter 3): (a) The CLUT markers visualized in the SI plane of the HSI color space. (b) The fully connected graph of the categorized CLUT markers that is subsequently filled to speedup the Fast Exact Euclidean Distance (FEED) transformation [254]. (c) The Weighted Distance Map (WDM), created using FEED transformation. (d) The chromatic-achromatic border, determined by a hill climbing algorithm, which can be described by Fourier functions [95].

(a) (b) (c) (d)

Figure 5.4: The processing scheme of the separation of the chromatic color categories in the hue-intensity (HI) plane, using human color categorization data (see Chapter 3) (note that the hue-axis is circular): (a) The CLUT markers visualized in the SI plane of the HSI color space. (b) The fully connected graphs of the categorized CLUT markers that are subsequently filled to speedup the Fast Exact Euclidean Distance (FEED) transformation [254]. (c) The labeled weighted distance map (WDM) created using FEED transformation. (d) The borders between the chromatic color categories, determined by a hill climbing algorithm, which can be described by Fourier functions [95].

5.5 Comparison of CLUT and segmented color space

Using the segmentation of the two 2D color planes, as described in the previous subsection, one of the 11 color categories is assigned to every point (i.e., color) in the whole 3D color space. The validation of this categorization method consisted of two tests and the analysis of their results: (i) categorization of non-fuzzy colors and (ii) categorization of fuzzy colors. The segmented color space is considered valid if and only if it categorizes the stimuli used in the experiments to the same categories as the subjects did.

The non-fuzzy colors are those colors that are categorized consistently to one color category by the participants of the experiments, as described in Chapter 3. The fuzzy colors are those colors categorized to two (or more) categories by at least 10 subjects. The fuzzy and non-fuzzy colors together make up the set of CLUT markers, derived from the experiments.

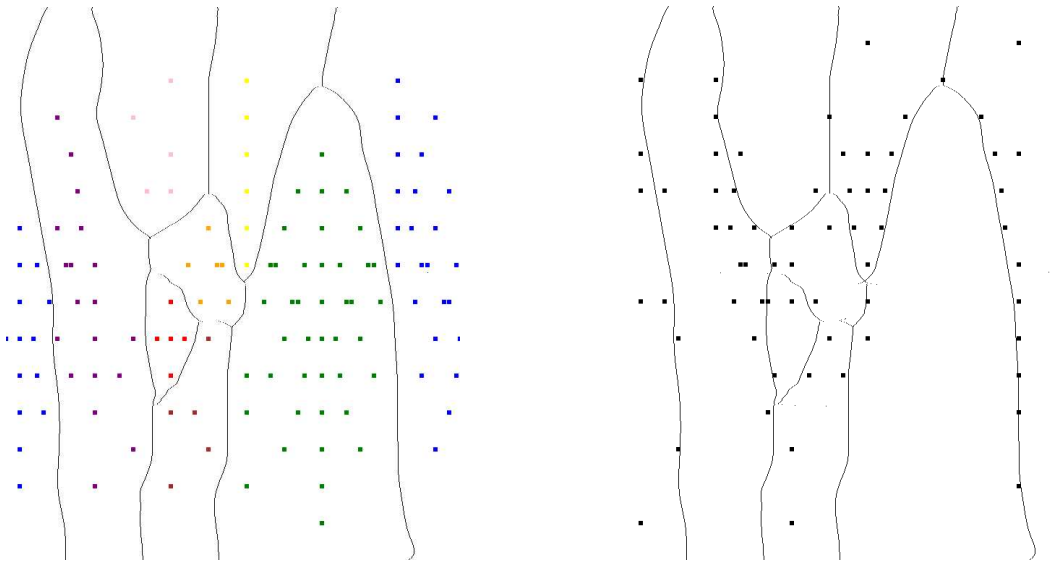
In Table 5.1, the 11 color categories are listed. The segmented color space has a 100% match with the experimental results. All non-fuzzy colors are categorized correctly. All fuzzy colors are mapped in one of the categories to which they were assigned to in the experiments.

Table 5.1: Colors and their neighbor colors in the segmented color space. The neighbor colors for each color category are found after analysis of the fuzziness of the experimental results.

	Purple	Pink	Orange	Red	Brown	Yellow	Green	Blue	Black	Gray	White
Purple	X	X	-	X	-	-	-	X	-	-	-
Pink	X	X	X	X	-	-	-	-	-	-	-
Orange	-	X	X	-	X	X	-	-	-	-	-
Red	X	-	X	X	X	-	-	-	-	-	-
Brown	-	-	X	X	X	-	-	-	-	-	-
Yellow	-	-	-	-	-	X	X	-	-	-	-
Green	-	-	-	-	-	X	X	X	-	-	-
Blue	X	-	-	-	-	-	-	X	-	-	-
Black	-	-	-	-	-	-	-	-	X	X	-
Gray	-	-	-	-	-	-	-	-	X	X	X
White	-	-	-	-	-	-	-	-	-	X	X

In Figure 5.5a, the non-fuzzy CLUT markers and the calculated borders of the chromatic categories have been plotted. In Figure 5.5b, the fuzzy CLUT markers in the plane used for chromatic segmentation, have been visualized.

Since the hue and intensity axis are able to describe non-overlapping clusters for chromatic categories, this 2D approximation was appropriate to segment the color space with FEED (using only hue and intensity values). Figure 5.5b shows that fuzzy data points are close to the calculated borders.



(a) (b)

Figure 5.5: The two dimensional HI plane with the calculated chromatic borders. (a) shows the non-fuzzy chromatic CLUT markers and (b) shows the fuzzy chromatic CLUT markers. Each dot represents a W3C web-safe color. In Appendix B the same Figure is shown as Figure B.4 with the non-fuzzy chromatic CLUT markers labeled with their focal color.

Please note that in both Figure 5.5a and in Figure 5.5b some data points are printed on the same spot in the HI plane. One should expect that the fuzzy and non-fuzzy set always represent different points in the HI plane. Since pre-filtering of the CLUT markers was done, removing all RGB values that were categorized to multiple categories, projection of the two data sets in a HI plane will (occasionally) show data points on the same spot.

5.6 Conclusion

We have explained our approach toward color analysis, which exploits human perception (i.e., the 11 color categories) instead of mere image processing techniques. The experimental data (i.e., markers for a Color LookUp Table), as described in Chapter 3, is used as input for a coarse color space segmentation process. The HSI color space is segmented, using two 2D projections of the HSI color space on which the recently developed Fast Exact Euclidean Distance (FEED) transform is applied. This resulted in an 11 color quantization scheme, a new color representation that is invariant for intensity changes.

The advantage of the color space segmentation method as proposed is threefold: (i) it yields perceptually intuitive results for humans, (ii) it has a low computational complexity, and (iii) it is easily adapted to other, application, tasks, and/or user dependent color categories, as will be shown in Chapter 8–13. Each of these three aspects are of extreme

importance [136] in image processing in general and especially in applications in the medical and cultural domain, where users are in interaction with the systems that analyze the images.

The use of the color space segmentation, as described in the current chapter, is illustrated in several research projects. Its results are published in over 20 articles (e.g., [31]). This thesis provides an overview of these projects and their reports.

In general, it is our belief that the combination of human perception and statistical image processing will improve the performance of data-mining systems that rely on color analysis. Moreover, such a combination can help us in bridging the semantic gap present in automated color analysis. From this perspective, a unique color space segmentation is proposed, which is both relevant and suitable for color analysis in the context of rapidly growing digital collections. Moreover, all means are provided to adapt it rapidly to a task, a specific application, or to a user profile [307].

In the remaining chapters, the segmented color space, as presented in this chapter, will be compared to other color quantization schemes for color-based image retrieval purposes. This will be done in various settings combined with other techniques. Moreover, it will be utilized for texture analysis (Chapters 10–13), image segmentation (Chapter 12), and shape extraction (Chapters 10 and 13). However, first a newly developed CBIR benchmark is introduced in which techniques and color quantization schemes can be compared to each other.

6

The Content-Based Image Retrieval (CBIR) Benchmark system

Abstract

Although the need for benchmarking content-based image retrieval systems is eminent, a lack of standards is still a major problem. This chapter introduces a CBIR benchmark (developed conform the W3C guidelines) that can be used online and offline. To guarantee real-time performance of the benchmark, the system uses cached matching results. Its standard database is a collection of 60,000 images from the Corel image database; however, the system can process any other image database. A set of four color quantization schemes (11, 64, 166, and 4096 bins) are included. Four distance measures are incorporated: the intersection distance, the Euclidean distance, the quadratic distance, and a newly developed distance measure, which takes into account intra bin statistics. Both other color quantization schemes and other distance measures can be easily included.

6.1 Introduction

In the previous chapter a new, human-centered, highly efficient color space segmentation was described, which can be applied as a color quantization scheme. However, to enable a comparison by users of this quantization scheme with other schemes, an evaluation of various methods had to be conducted. For the latter purpose, a CBIR benchmark is developed.

In the 90s, the first evaluations of color matching [138] and texture analysis [199] methods were published, followed by comparisons between algorithms for shape extraction and matching. In the last two years of this decade, the first evaluations of complete CBIR systems were presented [213, 214, 216]. These early evaluations were followed by a few others [151, 156] before the Benchathlon network [177] was initiated. Till then, as Gunter and Baretta [101] stated: "the performance of CBIR algorithms is usually measured on an isolated, well-tuned PC or workstation. In a real-world environment, however, the CBIR algorithms would only constitute a minor component among the many interacting components needed to facilitate a useful CBIR application; e.g., Web-based applications on the Internet." Hence, the Benchathlon network was founded to "develop an automated benchmark allowing the fair evaluation of any CBIR system" [177].

The aims of the Benchathlon Network can be found on its website [177] or in a series of articles [176, 186, 187, 188]. Next to the Benchathlon Network and Gunter and Baretta [101], Liu, Su, Li, Sun, and Zhang [161] conducted evaluations on CBIR systems. From various fields of research the need for benchmarking CBIR systems and their techniques was confirmed. Various evaluations of texture analysis techniques were presented [200, 257]. In Chapter 9–11 more texture evaluations are discussed and new texture evaluations are presented. From CBIR's fields of application the need for benchmarking CBIR systems also emerged; especially in medicine [68, 149, 185, 186, 188, 325] but also in the field of cultural heritage (e.g., museums and libraries) [296].

This chapter will discuss a CBIR benchmark used for various occasions, which is developed to facilitate judgment of retrieval results by users. It is used for the evaluation of (i) distance measures, throughout various chapters, (ii) color quantization methods (Chapter 7 and 8), (iii) texture analysis methods 11, and (iv) their combination (Chapter 12), which defined complete CBIR engines. The utilization of the benchmark throughout the research made sure that the human was constantly in the loop of the development of the CBIR techniques.

6.2 Benchmark architecture

The total benchmark system as developed in this project can be divided into two components, as is shown in Figure 6.1. The front-end of the system, contains the online (i.e., accessible through the WWW) user-interfacing module of the benchmark. Moreover, it includes offline (i.e., that are not run while the benchmark is used) modules that execute matching of color histograms and store matching results. The back-end, consists of a software module that handles the calculation and storage of color histograms.

Both components are connected to the same two databases: the image database and the histogram database. The image database contains 60,000 images of the Corel image database, which is one of the most frequently assessed image databases for CBIR research [186]. For texture analysis purposes, Müller et al. propose the MIT VisTex texture images database [178], included for research toward texture perception, see Chapter 11. The histogram database consists of several tables. For each quantization scheme in the benchmark, these tables contain the histogram for all images that occur in the image database.

6.2.1 Front-end

The online front-end module of the benchmark facilitates in user registration, user log-in, displaying matching results and storing user judgments. Additional features of this module are:

1. W3C HTML 4.01 Validation of benchmark [294], W3C Link Checking [309], and support of the "viewable with any browser campaign" [44].
2. Dynamic HTML: Size of the images are dynamically optimized to screen resolution.
3. Logging of the system configuration the participant is using; i.e., screen-size, settings of the video-card, operating system, and browser.
4. Automatic notification by e-mail of the start and end of the benchmark, to both participant and researcher.
5. Providing the possibility for a break at any time.

These features provide a general framework for multiple benchmark configurations, as described in Chapter 7 and 8 and used as the foundation of the Scientific User Interface Testing (SUIT) project [32].

The offline front end module of the benchmark is used to cache the matching results. In other words, the actual matching process is performed offline, prior to the benchmark. This saves processing time during the actual benchmark, which was preferred because then (i) we could guarantee a stable multi-user benchmark environment and (ii) the naive imple-

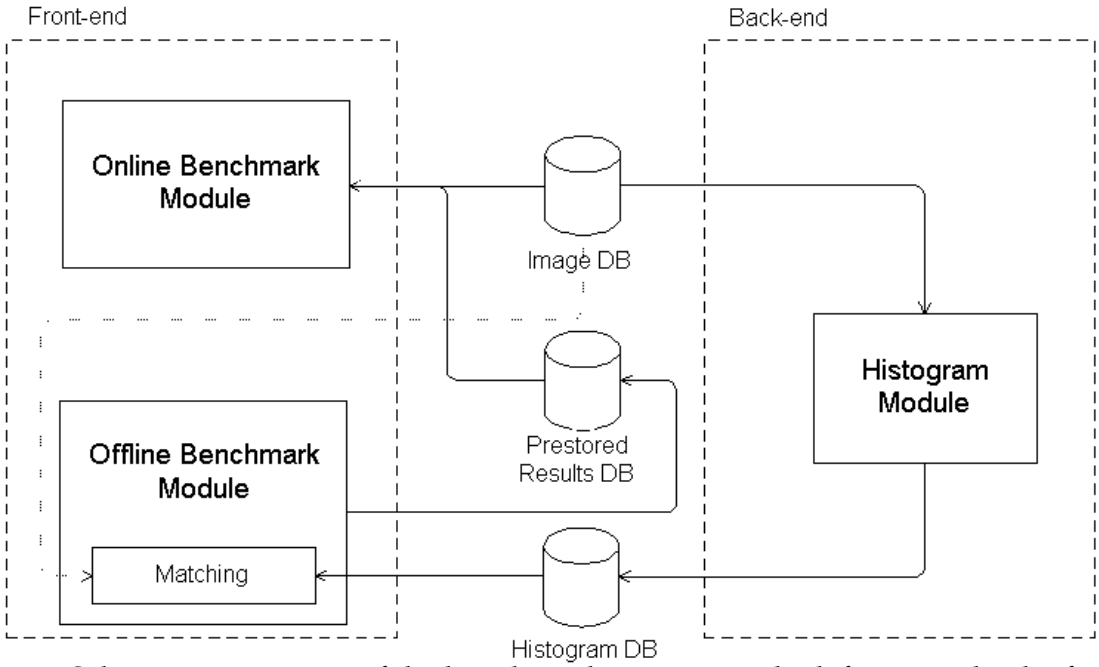


Figure 6.1: Schematic overview of the benchmark system. In the left rectangle, the front-end of the system is visualized, including the online and offline benchmark modules. In the right rectangle, the back-end of the system is visualized.

mentation of high dimensional quantization schemes, like RGB $16 \times 16 \times 16$ do not allow real-time image retrieval. Thus, for each quantization scheme in the benchmark the processing task was equal: retrieving the images from pre-stored lists.

The matching module controls the matching engine (software). Matching an image is done by retrieving histograms from the histogram database. Given a distance measure, which is provided by a parameter, the distance between the query histogram and the retrieved histograms is calculated. The distance measures used in the benchmark are described in Section 6.6. As soon as the ranking is known, the images are retrieved from the image database.

6.2.2 Back-end

The calculation and storing of histograms is done offline by the software module in the back-end component of the benchmark system. This module can be equipped with an arbitrary quantization scheme (e.g., see Chapter 5 and 6). Once the quantization schemes are defined, this software is able to calculate color histograms and store them in the appropriate histogram table. In order to execute, three parameters have to be provided:

File-listing: A text-file, which contains the disk location of the images, for which a color histogram has to be generated.

Quantization scheme: For each entry in the file-listing the module will create a color his-

togram using the quantization scheme that is specified by a parameter. Normally, the histogram is calculated by quantizing the color value of each pixel of an image, according to some color space. However, the 11 color categories require a somewhat different approach. To generate the color histogram, the pixels of an image have to be categorized by searching in the segmented HSI color space. This process is completely different than quantizing the axis of a color space, as was discussed in the previous chapter.

Table name: Once the calculation of a histogram is finished, the histogram database is accessed for storing it. The designated table is opened, a new histogram-ID is generated and the histogram is stored. The histogram database is discussed more thoroughly later on.

6.3 Image retrieval engine

In Figure 6.1, the matching module is visualized, encapsulated by the offline front-end of the benchmark. This program requires three parameters: (i) the index (ID) of the query image, (ii) the name of the histogram table denoting which quantization scheme is preferred, and (iii) the distance measure. In Section 6.6, the distance measures that are supported are defined. If preferred, this module can be easily equipped with additional distance measures.

Matching is done by calculating the histogram distance between the query-histogram and all other histograms in the histogram table. To limit memory use with respect to a very large histogram database, a linked list is updated during matching keeping track of the best 25 images. The output of the program is a list containing links to the top-25 images. In the benchmark setup, the offline benchmark module takes care of storing the output, which is printed on the command line. This engine is also used as part of an online retrieval test engine. In that setup, the output is the input for a program that generates `html` code for visualization of retrieval results.

6.4 Databases

The benchmark system contains three databases: (i) the image database, (ii) the histogram database, and (iii) the pre-stored results database. The default image database contains about 60,000 images of the Corel image database. The histogram database contains for each quantization scheme a unique histogram table. In general, these histogram tables are complete mappings of the image database. However, it is also possible to fill histogram tables with subsets, when only a specific category (subset or collection) of the image database is of interest. Each histogram table contains four fields of information, namely:

ID In this field, a unique ID is stored. This ID is currently used as a parameter for the matching software. In later developments, when the image database is very large (e.g., $>$ 1 million images), the ID can be used to express table relations. For example, the matching process can be made faster when indexing on the histogram table is done with use of pre-filtering results. These results are stored in a separate table using the ID of the histogram table as a reference.

Filename In this field the file-name and location of the image are stored.

Dimensions The size of the image is stored in this field. This information is used for histogram normalization during the matching process.

Histogram(n) This field contains the histogram of an image. Hence, for each quantization scheme the size of this field (n) is different (e.g., 11, 4096).

6.5 Result caching versus pre-storing of histograms

As mentioned before, the back-end facilitates in calculating and storing color histograms in a database, which is updated by an offline process. The matching engine, situated in the front-end, performs the matching process by retrieving histograms from this database. The processing time of the matching engine is thus not influenced by the calculation of histograms, but only by the retrieval of histograms and the calculation of histogram distances.

The benchmark system supports multiple image retrieval setups, with varying dimensionality. Therefore, it does not exploit online image retrieval, but rather uses cached matching results.

Given a query, the image retrieval process involves three stages that require processing time:

1. Calculation of the color histograms
2. Comparison of the histograms with a distance measure
3. Presenting matching results

A naive image retrieval system would perform these steps online. Pre-storing histograms allows to do the calculation of the first stage offline and speed up the online image retrieval process. By caching the matching results, the calculations of the second stage are also done offline. This caching mechanism guarantees a working space that is invariant to the complexity (dimensionality) of the image retrieval engine that is being benchmarked. Note, that this caching mechanism is only possible in a benchmark setup since the queries are known in advance. Hence, only the third stage: presentation of the images, has to be done online.

In Chapter 2, we have discussed the calculation of color histograms. The issue of pre-

senting the matching results will be briefly touched in the general discussion (Chapter 14). In the next section of this chapter, we will discuss the second stage as mentioned above: Comparison of color histograms, using a distance measure.

6.6 Distance measures

An image can be represented by a color histogram, defined by a color quantization scheme applied on a color space or on the 11 color categories (see also Chapter 2). In order to express the (dis)similarity of two histograms into a numeric value, a distance metric is used. In literature, a wide variety of distance measures can be found. For each distance measure the calculation method differs, leading to other estimations with respect to the similarity images represented by the histograms. As a consequence, the ranking of images, when a (query) image is compared to a set of images, will be different for each measure. Another difference between the distance measures is their computational complexity.

We will discuss three distance measures as applied in our research, starting with dissimilarity measures for feature vectors based upon the Minkowski metric: the intersection distance and the Euclidean distance. The Minkowski metric between two points $p = (x1, y1)$ and $q = (x2, y2)$ is defined as:

$$d^k(p,q) \;=\; \left(|x1 - y1|^k + |x2 - y2|^k\right)^{\frac{1}{k}}. \tag{6.1}$$

This metric can be adapted to compare histogram distance. The intersection distance was investigated for color image retrieval by Swain and Ballard [287]. The histogram distance, calculated per bin m, between a query image q and a target image t is denoted as:

$$D_i(q,t) = \sum_{m=0}^{M-1} |h_q[m] - h_t[m]|, \tag{6.2}$$

where M is the total number of bins, h_q is the normalized query histogram, and h_t is the normalized target histogram. We recognize $D_i(q,t)$ as the Minkowski form metric with k=1. The Euclidean distance is a Minkowski form with k=2:

$$D_e(q,t) = \sqrt{\sum_{m=0}^{M-1} (h_q(m) - h_t(m))^2}. \tag{6.3}$$

The distances (i.e., calculated Minkowski-form distance measures) only take account for the correspondence between each histogram bin (see Figure 6.2a) and do not make use of information across bins. This issue has been recognized in histogram matching. As a result, quadratic distance is proposed to take similarity across dimensions into account (see Figure 6.2b). It has been reported to provide more desirable result than only matching be-

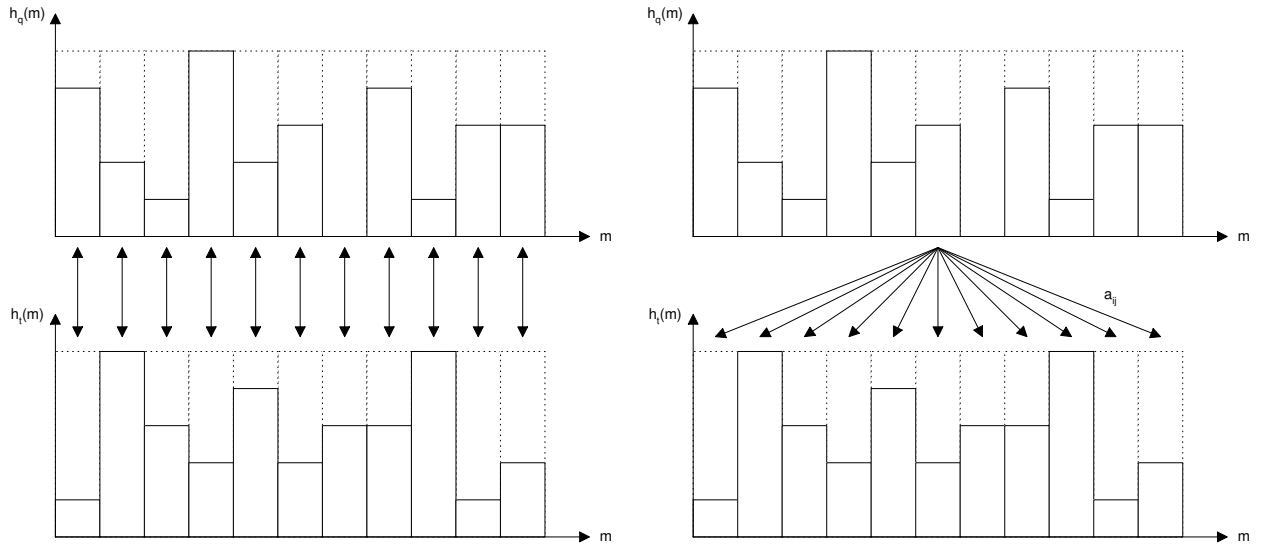


Figure 6.2: (a) Minkowski-form distance metrics compare only similar bins between histograms. (b) Quadratic-form distance metrics compare multiple bins between histograms using similarity matrix $A = [a_{ij}]$.

tween similar bins of the color histograms [242]. However, since the histogram quadratic distance computes the cross similarity between colors, it is computationally expensive. The quadratic-form distance between two feature vectors q and t is given by:

$$D_q(q,t) = (h_q - h_t)^T \mathbf{A}(h_q - h_t), \tag{6.4}$$

where $\mathbf{A} = [a_{ij}]$ is a similarity matrix. a_{ij} denotes the similarity between elements with indexes i and j. Please note, that h_q and h_t are denoted as vectors.

The three distance measures as presented in this section are those which were applied in our research. However, a broad range of alternative distance (and similarity) metrics exists; e.g., the binary set quadratic distance and the Mahalanobis distance. For thorough descriptions of distance and similarity measures, we refer to Smith [273].

Note that the distance measures used, require the normalization of histograms. This can be achieved either by normalizing all images to an arbitrary size before they are processed or by normalizing the histogram resulted from the processed image. For the latter, we introduce the following class of normalizations:

$$h^k = \frac{h}{(\sum_{m=0}^{M-1} |h[m]|^k)^{1/k}}. \tag{6.5}$$

Moreover, the number of bins must be the same and the color information coded in the bins, which are compared, must be equal for all images under investigation.

6.7 Conclusions

The benchmarking framework we developed, provides means to test and compare (parts of) CBIR engines. To ensure fast performance, we executed the benchmark with retrieval results that were already stored (cached). So, users can judge retrieval results, independent of retrieval speed. This is conform the advice of Leung and Ip [156] who stated that "a central aim for the benchmark is to test the ability to retrieve images by content rather than the speed of retrieval." In addition, Puzicha and Rubner [216] stated: "Performance comparisons should account for the variety of parameters that can affect the behavior of each measure. ... A fair comparison in the face of this variability can be achieved by giving every measure the best possible chance to perform well."

In the next two chapters, two benchmarks are presented. The first benchmark was developed to evaluate both low and high dimensional retrieval engines. However, exhaustive evaluation of distance measures such as done by Zhang and Lu [325], was not our intention. In order to restrict the size of the first benchmark, we selected a simple (i.e., the intersection distance [287]) and an advanced distance measure (the quadratic distance [193]), suited for respectively low and high dimensional schemes. The first benchmark incorporates four quantization schemes (i.e., 11, 64, 166, and 4096 bins). Based on the results of the first benchmark, the second benchmark uses only two quantization schemes (i.e., 11 and 4096 bins) and combines these with the intersection measure and with a newly developed distance measure, which utilized intra-bin statistics. In Chapters 11 and 12 the benchmark is also used for judging respectively texture analysis techniques and object-based image retrieval engines.

7

The utilization of human color categorization for content-based image retrieval

Abstract

We present the concept of intelligent Content-Based Image Retrieval (iCBIR), which incorporates knowledge concerning human cognition in system development. The present research focuses on the utilization of color categories (or focal colors) for CBIR purposes, in particularly considered to be useful for query-by-heart purposes. The research presented in this chapter explores its potential use for query-by-example purposes. Their use was validated for the field of CBIR in two experiments (26 subjects; stimuli: 4 times the 216 W3C web-safe colors) and one question ("mention ten colors"). Based on the experimental results a Color LookUp Table (CLUT) was defined. This CLUT was used to segment the HSI color space into the 11 color categories. With that a new color quantization method was introduced making a 11 bin color histogram configuration possible. This was compared with three other histogram configurations of 64, 166, and 4096 bins. Combined with the intersection and the quadratic distance measure we defined seven color matching systems. An experimentally founded benchmark for CBIR systems was implemented (1680 queries were performed measuring relevance and satisfaction). The 11 bin histogram configuration did have an average performance. A promising result since it was a naive implementation and is still a topic of development.

This chapter is almost identical to:

Broek, E. L. van den, Kisters, P. M. F., and Vuurpijl, L. G. (2004). The utilization of human color categorization for content-based image retrieval. *Proceedings of SPIE (Human Vision and Electronic Imaging IX), 5292*, 351-362.

7.1 Introduction

Digital media are rapidly replacing their analog counterparts. Less than 10 years ago a digital photo camera was solely used in professional environments [129]. In contrast, nowadays many home users own a digital photo camera. This development is accompanied by (i) the increasing amount of images present on the Internet, (ii) the availability of the Internet for an increasing number of people, and (iii) a decline in digital storage costs.

As a result, the need for browsing image collections emerged. This development gave birth to a new field within *Information Retrieval (IR)*: image retrieval. When images are part of a web-page or when images are textually annotated in another form, IR-techniques can be utilized. However, how to search for images that are not properly annotated? We will first discuss quantitative followed by qualitative arguments, that point out the relevance of *Content-Based Image Retrieval (CBIR)* . CBIR uses features of the image itself (i.e., color, texture, shape, and spatial characteristics), which enables us to search for images that are not textually annotated.

Murray [189] determined in his paper "Sizing the Internet" on July 10, 2000 that 2.1 billion unique pages were present on the Internet. He further states that "Internet growth is accelerating, indicating that the Internet has not yet reached its highest growth period.". Currently, estimates of the number of unique pages range from over 50 million [191] up to over 12 billion [123]. In addition, Murray determined the average number of images present on a page to be 14.38. One year later Kanungo et al. [137] drew a sample of Internet (consisting of 862 pages) and determined the average number of images per page as being 21.07. Unfortunately, neither of these papers report their definition of an image. The latter is of importance since one can make a distinction between images (e.g., cartoons and photos) and web graphics (i.e., backgrounds, bullets, arrows, and dividers). Furthermore, the size of the "invisible web" (i.e., databases available through websites) was not taken into account. From the previous facts can be derived that between 720 million and 250 billion images are present on the Internet. Due to a lack of statistical data, we can not make an estimation of the two other sources of images: the "invisible Internet' and home users' private image collections. However, it is safe to say that these two latter sources of images, will increase the number of images substantially.

Next to the quantitative argument as discussed above, a qualitative argument can be made that illustrates the importance of CBIR. Let P be a square of pixels. P either consist of characters c or is an image i. Let i be a graphic itemizing bullet, typically of size 8^2 to 26^2 pixels. Let c be the word "the". Using a standard font-size the word "the" needs a square of 17^2 pixels, which equals the average size of a bullet. However, a graphic itemizing

bullet can, for example, resemble "the footprint of a bear"[1] using as much pixels but having a much richer semantic content. So, the saying "a picture is worth more than a thousand words" holds when considering the semantics that can be expressed per area.

Based on these considerations it can be concluded that CBIR is of significant importance for IR in general, for retrieval of unannotated images in image databases, and for home users that manage their own image (e.g., photo) collections. However, as Smeulders et al. [270] noted in 2000 "CBIR is at the end of its early years" and is certainly not the answer to all problems. To mention a few, CBIR-engines are not capable of searching beyond a closed domain, are computationally too expensive, have a low retrieval performance, and do not yield results that match the needs of the user. Therefore, the CBIR techniques used are still subject of development.

Due to the large differences between users' search strategies, even interactive user-adaptable CBIR-engines have been proposed [87, 92]. Such an approach is as useful as it is complex. We will attempt to find out whether such an approach is needed at this moment in CBIR development.

Our approach to improve the performance of CBIR systems is through the utilization of knowledge concerning human cognitive capabilities. In our research the distinction is made between *query-by-memory* and *query-by-example* , each requiring cognitive processes. With query-by-memory the user has to define image features by memory, whereas in case of query-by-example an example image is supposed to be present. A CBIR engine uses features such as shape, spatial characteristics, texture, and color to explore the image content. The current research will focus on the latter feature: color. It will be shown in this chapter that human color categories can be utilized for CBIR techniques.

7.1.1 Query-by-Example versus Query-by-Heart

Most CBIR-engines distinguish two forms of querying, in which the user uses either an example image (query-by-example) or defines features by heart, such as: shape, color, texture, and spatial characteristics (query-by-heart). In the latter case, we are especially interested in the use of the feature color. In the remaining part of this article we therefore define query-by-heart as query-by-heart utilizing color. At the foundation of both query-by-example and query-by-heart, lies a cognitive process, respectively color discrimination and color memory. Let us illustrate the importance of the distinction between query-by-example and query-by-heart by a simple example. Imagine a user wants to find images of brown horses.

Suppose the user possesses one such image and uses it to query-by-example. Images found will be matched to the example image by the CBIR engine. The resulting images are

[1] Text and bullet are present on: http://www.w3schools.com/graphics/graphics_bullets.asp [accessed on July 31, 2005]

presented to the user. The user compares all retrieved images with his own image and with each other. This comparison we call the process of color discrimination. So, in this process the colors are (directly) compared to each other.

In the case of query-by-heart the user is required to retrieve the color brown from memory. Probably, this will not be one particular color, but rather a fuzzy notion of some set of colors: a color category, based on color memory. Each of the elements of this brown set (or category) are acceptable colors. There is no need for several types of brown. Providing the keyword "brown" or pressing a button resembling the fuzzy set brown is sufficient.

In both forms of querying the CBIR-system can use a Color Look-Up Table (CLUT) for the determination of the elements of this set, described by R, G, and B-values. The set is fuzzy due to the several influences on the color (of the object of interest), such as the color of the surrounding and the semantic context in which the object is present.

However, it is clear that a distinction should be made between color categorization by discrimination and color categorization by memory. An important distinction because humans are capable of discriminating millions of colors but when asked to categorize them by memory, they use a small set colors: focal colors or color categories [16, 93, 232]. Despite the fact that the importance of such a distinction is evident, this differentiation is not made in CBIR-systems.

We propose to use the 11 color categories for query-by-heart purposes in CBIR. For this purpose the front end of a CBIR engine was already extended with an eleven color pallet, as described in [42]. The 11 color matching engine perfectly fits this interface. However, we wanted to explore the use of the 11 color categories further and extend their use to query-by-example. But before this was done an endeavor was made toward experimental evidence for the 11 color categories.

7.2 The benchmark

In order to assess the validity of our approach for image retrieval, we have made a first comparison study with the three color matching algorithms described above. The field of IR provides two metrics for estimating retrieval effectiveness: recall and precision. Recall signifies the relevant images in the database that are retrieved in response to the query. Precision is the proportion of the retrieved images that are relevant to the query.

$$\text{recall} = \frac{\#\text{relevant retrieved}}{\#\text{relevant}} \qquad \text{precision} = \frac{\#\text{relevant retrieved}}{\#\text{retrieved}} \tag{7.1}$$

The key issue is to determine which images are relevant. Merriam-Webster's dictionary [181] defines relevance as " *the ability (as of an information retrieval system) to retrieve material that satisfies the needs of the user* ". So, relevance concerns the satisfaction of the user.

The judgment of users is the only way to determine the recall and precision of the matching algorithms [116].

As we are unable to a priori approximate the number of relevant images for a given query, it is not possible to determine the recall of the systems. We can only examine their precision. The number of retrieved images follows from the retrieval and is fixed to 15 for this experiment. So, it is required to know the number of relevant retrieved images, for which the experiments described in this section are used.

7.2.1 Histogram configurations

Four histogram configurations were used (11, 64, 166, and 4096 bins), each having their own quantization method. For the histogram configuration using 11 bins a quantization method was used based on the proposed segmented HSI color space. The configuration containing 64 bins is inspired by the PicHunter [63] image retrieval engine, which uses a HSV($4\times4\times4$) quantization method. The quantization method used for the 166 bins is similar to the approach described in [274]. We call the configuration HSV($18 \times 3 \times 3$)+4, meaning that the quantization was performed for 18 hues, 3 saturation, 3 values, and 4 achromatics (representing the central rod in the HSV color space). The last histogram configuration is the QBIC configuration using 4096 bins [85, 193]. The quantization method used for this configuration is RGB($16 \times 16 \times 16$). This (computational heavy) configuration is picked to show the insignificance of the color space (used for quantization) when a large number of bins is used. Please note that the color histogram matching methods described in the previous Section have been implemented for this experiment and that no efforts have been made to exactly copy the optimized matching algorithms of the PicHunter, QBIC, and the system described by [274].

7.2.2 Distance measures

Two histogram matching functions are used in the current setup of the benchmark: the histogram intersection distance and the quadratic distance. For other histogram matching functions we refer to works of Gonzales [95] and Puzicha [216]. We have chosen for these two measures because: (i) the intersection distance is one of the most used and widely excepted measures, (ii) the quadratic distance is reported as performing good [193], and (iii) we had to limit the number of measures since our focus lies on quantization and a benchmark should be workable. Exhaustive testing of all distance measures was therefore not conducted.

Swain's [287] color-indexing algorithm identifies an object by comparing its colors to the colors of each of the potential target objects. This is done by matching the color his-

tograms of the images via their histogram intersection; see also the previous chapter. The quadratic distance measures is used in QBIC [85]; see also the previous chapter. Since it is computationally expensive in its naive implementation optimizations are proposed, as described in [102].

7.2.3 Design

For the benchmark two distance measures are chosen: the intersection distance (see Equation 6.2) and the quadratic distance (see Equation 6.4). We have used the four histograms, consisting of respectively 11, 64, 166, and 4096 bins (as described in 7.2.1. Each distance measure was applied on each number of bins, with one exception. The combination of 4096 bins with the quadratic distance measure was found computationally too expensive to use[102]. So, as denoted in Table 7.1, in total seven systems that are compared in this benchmark.

Table 7.1: The seven engines incorporated in the CBIR-benchmark, defined by their color quantization scheme (i.e., 11, 64, 166, or 4096 bins) and the distance measure applied; i.e., the intersection distance (ID) or the quadratic distance (QD).

	11 bins	64 bins	166 bins	4096 bins
Intersection distance	ID-11	ID-64	ID-166	ID-4096
Quadratic distance	QD-11	QD-64	QD-166	

For each system, 20 query results had to be judged by human subjects, making a total of 140 per subject. Each set of 140 queries was fully randomized, to control for influence of order. Normally such retrieval results are presented in their ranked order. However, if this would have been done in the experiment the subjects would be biased to the first retrieval results after a few queries. Therefore, the ranking of the retrieved images is presented in random order.

Each query resulted in 15 retrieved images, presented in a 5×3 matrix. On the left side of this matrix the query image was shown. The layout (4:3) and the size of the images were chosen in such a way that the complete retrieval result was viewable and no scrolling was necessary (see Figure 7.1).

7.2.4 Subjects, instructions and data gathering

12 subjects, both men and women in the age of 20-60, participated in the benchmark, making a total of 1680 query-results (one of them did not finish the experiment). The subjects were asked to judge the retrieved images solely based on the color distribution hereby ignoring the spatial distribution of the colors. It was emphasized that semantics, shape, etc. should

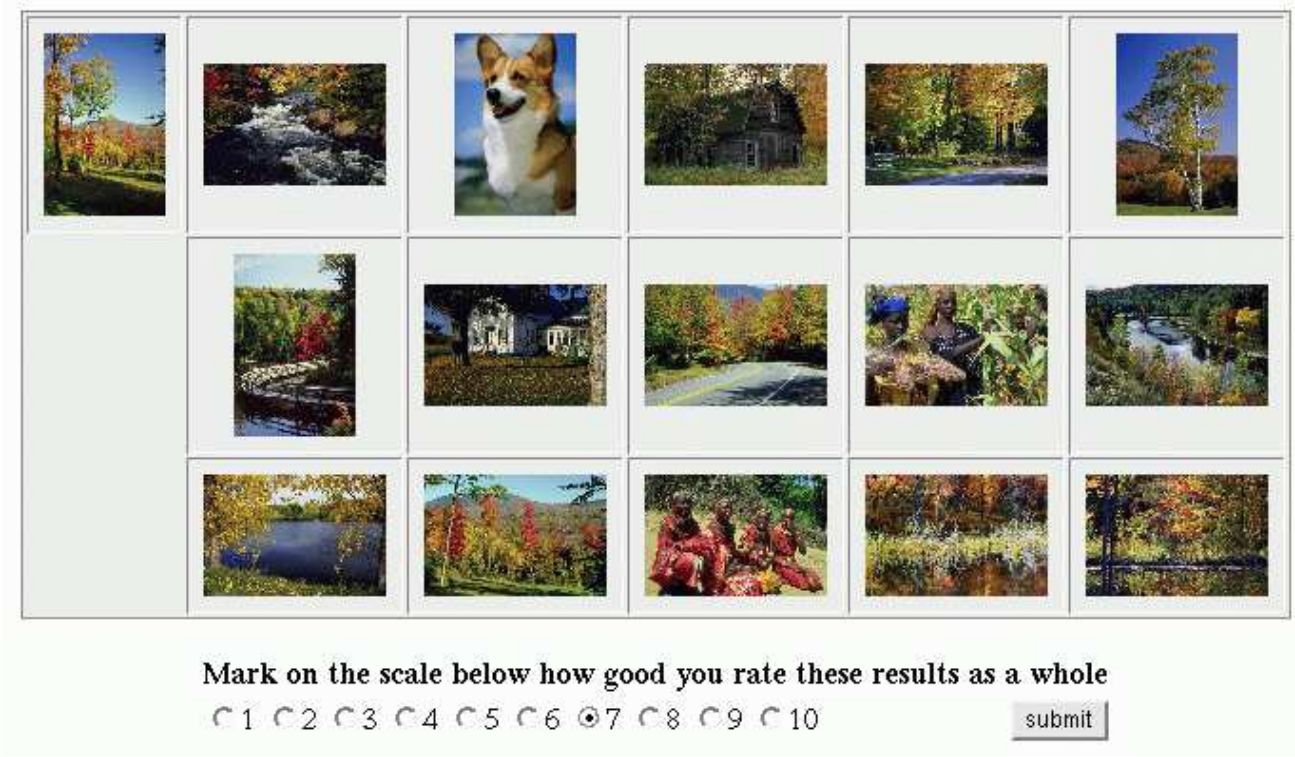


Figure 7.1: The interface of a query such as was presented to the subjects. They were asked to select the best matching images and to rate their satisfaction. See Figure B.6 in Appendix B for a large full color print of this screendump.

not influence their judgment. The judgment of the subjects was two-fold. On the one hand they were asked to mark the images that they judged as relevant. On the other hand, they were asked to indicate their overall satisfaction with the retrieved results on a scale from 1 to 10 (see Figure 7.1).

We recorded for each query of each participant: the image ID, the query number, the distance measure used, the number of bins used, satisfaction rate, and images judged as relevant.

Both the number of selected images and the rating for each query were normalized per person. This was necessary since the range of the number of selected images as well as the rating of satisfaction varied strongly between subjects. The normalized values were used for the analysis. How this normalization was done is defined in the next section.

7.2.5 Normalization of the data

The participants' strategies for selection and rating of the retrieved images varies enormously. On behalf of the analysis, a normalization of the scores was applied, for each participant separately.

Normalizing a range of scores takes the maximum and minimum score possible and the maximum and minimum score provided by the participant into account. Such a transformation is defined by:

$$S_n = a \cdot S_p + b, \tag{7.2}$$

where S_n is the normalized score, S_p is the score provided by the participant, and a and b are defined as:

$$a = \frac{\max - \min}{\max_p - \min_p} \qquad b = \max - a \cdot \max_p \tag{7.3}$$

with $\max$ and $\min$ being respectively the maximum and minimum score possible and $\max_p$ and $\min_p$ being the maximum and minimum score provided by the participant. Note that where S_p is an integer, the normalized score S_n is a real number, since both a and b are real numbers. However, this is not a problem for further analysis of the data.

7.2.6 Results

We have analyzed the data using six one-way ANalyses Of VAriance (ANOVA). The systems were compared as well as their compounds: their histogram configuration (11, 64, 166, and 4096 bins) and their distance measure (quadratic distance QD and the intersection distance ID). This was done for the number of images indicated as relevantly retrieved, by the users and for the overall rated satisfaction of the query-result, as indicated by the users. See also Table 7.1, which provides an overview of all engines and their compounds. Table 7.2 provides the mean number of images selected and Table 7.3 provides the mean rated satisfaction for each of these engines.

For both the relevant retrieved images as the satisfaction rate, strong differences were found between the seven systems (resp. F(6,1673) = 10.39, p $<$ .001 and F(6,1673) = 12.72, p $<$ .001). Nevertheless, for both the relevant retrieved images and the rated satisfaction, the systems could be clustered in three groups, using Duncan's multiple range post hoc test [81, 107] ($p < .05$). This was done by the construction of homogeneous subsets (i.e., clusters), assuming equal variances. In Tables 7.2 and 7.3 the three groups are denoted by italic, bold, and underlined fonts.

A more detailed analysis, revealed a clear influence of both the distance measure (F(1,1678) = 38.09, p $<$ .001) and the number of bins (F(3,1676) = 12.75, p $<$ .001), on the number of relevant images. The same results were shown on the satisfaction for both the distance measure (F(1,1678) = 45.74, p $<$ .001) and the number of bins (F(3,1676) = 15.73, p $<$ .001).

In order to determine the variability between subjects in their judgments two additional one-way ANOVAs were done. Their outcome was that the subjects differ in a large

Table 7.2: The seven engines incorporated in the CBIR-benchmark, defined by their color quantization scheme (i.e., 11, 64, 166, or 4096 bins) and the distance measure applied; i.e., the intersection distance (ID) or the quadratic distance (QD). For each engine, the average number of images denoted (i.e., selected) as being correct is provided. Moreover, the engines were clustered by performance using the Duncan's multiple range test ($p < .05$). Three clusters were identified, denoted by italics, bold and underlined fonts.

	11	64	166	4096
ID	*3.78*	**4.41**	**4.65**	5.14
QD	*3.44*	*3.56*	*3.61*	

Table 7.3: The seven engines incorporated in the CBIR-benchmark, defined by their color quantization scheme (i.e., 11, 64, 166, or 4096 bins) and the distance measure applied; i.e., the intersection distance (ID) or the quadratic distance (QD). For each engine, the average rating of satisfaction are provided. Moreover, the engines were clustered by performance using the Duncan's multiple range test ($p < .05$). Three clusters were identified, denoted by italics, bold and underlined fonts.

	11	64	166	4096
ID	*4.93*	**5.47**	**5.60**	6.06
QD	*4.71*	*4.72*	*4.80*	

extent on both their satisfaction (F(11,1668) = 38.77, p $<$ .001) and the as relevant judged images (F(11,1668) = 39.03, p $<$ .001). For satisfaction we identified five groups (i.e, clusters) of subjects and for the number of relevant images we were able to identify seven groups (i.e, clusters) of subjects, again using Duncan's multiple range post hoc test [81, 107] ($p < .05$). Since 12 subjects participated, this illustrates the enormous inter personal differences in rated satisfaction and in the judgment of when an image is relevantly retrieved.

7.3 Discussion

The new color quantization scheme (or segmentation), introduced in the previous chapter, was compared with histograms consisting of 64, 166, and 4096 bins. This was done using two distance measures: the intersection distance and the quadratic distance. The seven resulting systems were tested in a benchmark.

The system that combined the 11 bin histogram with the intersection distance measure performed better than all systems using quadratic measures, but it performed not as good as systems using a stronger quantization of color space (i.e., used histograms with resp. 64, 166, 4096 bins) combined with the intersection distance measure. So, our naive implementation of the 11 bin concept should be boosted in order to be comparable with systems using histograms with more bins.

A few explanations can be given for the lack of performance. Since, we have used an extreme weak quantization relatively, much performance can be gained by incorporating statistical techniques, such as within-bin color distributions. Moreover, the 11 color category matching approach was initially developed for query-by-heart purposes, where with the benchmark it was tested for query-by-example purposes.

However, an advantage of the 11 bin approach is its low computational complexity. On this issue, the 11 bin concept outperforms the other histograms by far. Taking in consideration that the latter is of extreme importance [136] in the field of CBIR, the results were very promising. In the next chapter, an enhanced 11 color categories approach is introduced, accompanied by a matching, new similarity measure.

8

Content-Based Image Retrieval benchmarking: Utilizing color categories and color distributions

Abstract

In this chapter, a new weighted similarity function was introduced. It exploits within bin statistics, describing the distribution of color within a bin. In addition, a new CBIR benchmark was successfully used to evaluate both new techniques. Based on the 4050 queries judged by the users, the 11 bin color quantization proved to be useful for CBIR purposes. Moreover, the new weighted similarity function significantly improved retrieval performance, according to the users.

This chapter is a compressed version of:

Broek, E. L. van den, Kisters, P. M. F., and Vuurpijl, L. G. (2005). Content-Based Image Retrieval Benchmarking: Utilizing Color Categories and Color Distributions. *Journal of Imaging Science and Technology, 49(3)*, 293–301.

The organization of this chapter is as follows. We start with the introduction of the enhanced 11 bin color quantization in Section 8.1. In Section 8.2, the newly developed accompanying similarity function for query-by-example is defined. Both the color space segmentation, which resulted in a color quantization scheme for CBIR, and the new similarity function are applied in a benchmark. The three CBIR systems used in the benchmark, the method of research, the results and the discussion of the results can be found in Section 8.4.

8.1 Enhanced 11 bin color quantization

The 11 bin quantization of color space was originally developed for query-by-memory (see Chapter 5). So, the user has to rely on his limited color memory when judging the retrieved images. For the query-by-example paradigm, the drastic reduction of color information to 11 color categories (or bins) is coarse, as was shown in Chapter 5.

However, query-by-example is of importance for CBIR since it has two advantages compared to query-by-memory: (i) It requires a minimal effort of the user and (ii) It is the most wieldy paradigm since all possible features (color, texture, shape, and spatial information) can be analyzed. In a query-by-memory the latter is hard and partially impossible. For example, users experience it as difficult to sketch a shape [308] and are not capable of defining complex textures. Since query-by-example is such an important paradigm for CBIR, we should aim to adopt the 11 bin quantization scheme to the query-by-example paradigm.

We will now explain that instead of adopting a more precise quantization scheme, the notion of the 11 color categories should be preserved. However, a higher precision is needed for the 11 bin quantization scheme.

In the previous chapter, the 166 bin quantization $(18 \times 3 \times 3)$ of HSV color space was not judged as performing significantly better in query-by-example than the 64 bin quantization $(4 \times 4 \times 4)$ of HSV color space. This despite the fact that the 166 bin quantization is 2.6 times more precise than the 64 bin quantization. Hence, a more precise quantization scheme is not a guarantee for success. In addition, in the same study the 11 bin color quantization performed as well as the more precise, 64 and 166 bin quantizations. So, the 11 bin quantization can be considered as an extremely efficient color quantization scheme.

The success of the 11 bin color quantization scheme can be explained by its origin: human color categorization, where the 64 and 166 bin quantization schemes naively segmented each of the three axes of HSV color space into equal segments.

One way to extend the 11 color histogram would be to divide each color category in a number of segments, for example, relative to the size of the area each category consumes in the HSI color space. However, with such an approach only the number of pixels present in a bin are taken into account; color variations within bins are ignored. Therefore, we chose to

incorporate statistical information that describes the distribution of pixels within each bin.

Such an approach is only useful if a segment of color space represented by a bin is perceptually intuitive for the users. The naive 64, 166, and 4096 bin quantizations as used in previous chapter are not perceptually intuitive for users. For these quantization schemes, the incorporation of statistical data would not make sense, since the sections in the color space contain heterogeneous data (i.e., colors).

Since the statistical values can be precomputed and stored, these can be represented as a vector of size $n * a$ where n is the number of bins and a is the number of statistical values per bin. This representation is similar to the vector-representation of a histogram. Therefore, each statistical value can be represented as a virtual bin. Therefore, such an approach is relatively cheap compared to a more precise quantization.

In the next section, we will describe the within bin statistical information and how it is used as a similarity measure.

8.2 Similarity function using within bin statistics

8.2.1 The intersection similarity measure

A distance measure calculates the distance between two histograms. A distance of zero represents a perfect match. We use the histogram intersection distance (D) of Swain and Ballard [287] between a query image(q) and a target image (t):

$$D_{q,t} = \sum_{m=0}^{M-1} \mid h_q(m) - h_t(m) \mid, \tag{8.1}$$

where M is the total number of bins, h_q is the normalized query histogram, and h_t is the normalized target histogram.

When combining distance measures (by multiplying them), a single perfect match would result in a perfect match for the total combination. However, this is an unwanted situation since one would expect a perfect match if and only if all distance measures indicate a perfect match. Therefore, the similarity (i.e., similarity = 1 - distance) for each variable is calculated, instead of its distance.

In order to determine the intersection similarity (S) we adapt Equation 8.1 to give:

$$S_{q,t} = \sum_{m=0}^{M-1} 1 - \mid h_q(m) - h_t(m) \mid . \tag{8.2}$$

8.2.2 Extension of the intersection measure

Based on Equation 8.2 a new distance measure is developed, incorporating statistical information of each color category separately. In histogram matching only the magnitudes of the bins are of importance (i.e., the number of pixels assigned to each bin). However, for our new distance measure we will use five values next to the amount of pixels in the bins.

These values are stored in a *color bucket b* , assigned to every color category (or quantized color space segment):

$$\begin{cases} x_1(b) = \#(b) & \text{(i.e., the amount of pixels in bucket } b\text{; the original histogram value } h) \\ x_2(b) = \mu H(b) & \text{(i.e., the mean hue } H \text{ of bucket } b) \\ x_3(b) = \mu S(b) & \text{(i.e., the mean saturation } S \text{ of bucket } b) \\ x_4(b) = \sigma H(b) & \text{(i.e., the } SD \text{ of the hue values } H \text{ in bucket } b) \\ x_5(b) = \sigma S(b) & \text{(i.e., the } SD \text{ of the saturation values } S \text{ in bucket } b) \\ x_6(b) = \sigma I(b) & \text{(i.e., the } SD \text{ of the intensity values } I \text{ in bucket } b), \end{cases}$$

where $x_i(b)$ denotes value i of color bucket b of either query image q: $q_i(b)$ or of target image t: $t_i(b)$. SD is the abbreviation of Standard Deviation.

For each pair q_i and t_i (with $i \epsilon \{1, 6\}$), of each bucket b the similarity S_{q_i,t_i} is determined, as follows:

$$S_{q_i,t_i}(b) = 1 - \mid q_i(b) - t_i(b) \mid, \tag{8.3}$$

where the range of S_{q_i,t_i} is [0,1].

For the buckets representing the achromatic color categories, no values were calculated for the hue and saturation axis. The achromatic color categories are situated in the central rod of the HSI model. Hue values are represented by the angle around this rod (indicating the basic color). Saturation values refer to the distance of a point to the central rod. The larger the distance, the stronger the color information of a certain hue, is present.

Achromatic colors show very small values for saturation, regardless of their hue angle. Therefore, when referring to achromatic categories, statistical information about the hue and saturation axis does not contribute to the precision of the search algorithm and is, therefore, ignored in the algorithm. To achieve the latter, by definition $\mu H(b) = \mu S(b) = \sigma H(b) = \sigma S(b) = 0$ for buckets b representing achromatic colors. This results in $S_{q_2,t_2}(b) = S_{q_3,t_3}(b) = S_{q_4,t_4}(b) = S_{q_5,t_5}(b) = 1$.

In addition, note that the mean values for the third axis of the HSI color-space, the intensity axis, are not used for similarity calculation. With the exclusion of the mean intensity for each bucket, the similarity measure is intensity invariant, which enables generalization in matching. However, this advantage can, for example, become a disadvantage in a setting

where solely color levels are compared.

Now that all values of a bucket are described, the total similarity for each color bucket b (i.e., a quantized color category) can be defined as:

$$S_{q,t}(b) = S_{q_1,t_1}(b) \cdot S_{q_2,t_2}(b) \cdot S_{q_3,t_3}(b) \cdot S_{q_4,t_4}(b) \cdot S_{q_5,t_5}(b) \cdot S_{q_6,t_6}(b) \tag{8.4}$$

In addition to the statistical information, extra histogram information is used for determining the similarity. For each color bucket b of the query image q a weight-factor $W_q(b)$ is calculated. The weight is proportional to the amount of pixels in a bucket. So, the most dominant color category of the query image, having the most pixels, has the largest weight. The reason to add such a weight is twofold. First, small buckets that represent a relative small amount of pixels do not disturb the similarity calculation. Second, empty buckets do not join the similarity calculation, because their weight is zero.

$$W_q(b) = \frac{q_1(b)}{\sum_{i=0}^{B-1} q_1(i)} \tag{8.5}$$

where B is the total number of color buckets. Further, please note that for a normalized histogram, as is the case in the present research, Equation 8.5 can be rewritten as:

$$W_q(b) = q_1(b). \tag{8.6}$$

The total image similarity is then defined as:

$$S_{q,t} = \sum_{b=0}^{B-1} S_{q,t}(b) \cdot W_q(b). \tag{8.7}$$

A technical advantage of this similarity measure, which is incorporated in the 11 bin matching engine, is that it can be used or can be ignored when matching. The matching performance in a query-by-example setting will benefit from the additional information. For the query-by-memory paradigm the same engine can be used, but when preferred, the statistical information can be excluded.

8.2.3 Computational complexity

Each statistical value can be regarded as a virtual bin. For all 11 bins the standard deviation of the intensity (σI) is determined. In addition, for the 8 chromatic colors the mean hue (μH), the mean saturation (μS), the standard deviation of the hue (σH), and the standard deviation of the saturation (σS) are determined. So, for the enhanced 11 bin configuration a total of $11 + 8 \cdot 4 = 43$ virtual bins are added. Hence, the computational complexity of the enhanced 11 bin configuration is equal to that of a $11 + 43 = 54$ bin histogram.

8.3 Theoretical considerations: From color concepts to color perception

The original feature vector, based on the 11 color categories, was originally developed for the query-by-memory paradigm. With respect to the query-by-example paradigm, we developed an extended color feature vector, which contains statistical color values for each color category. This approach has four major advantages:

1. A rich source of additional information (i.e., 43 color features) is available.
2. The human-centered color space segmentation is utilized.
3. The CBIR engine is still computationally inexpensive [136] (i.e., only 54 dimensions).
4. The additional information is based on color features that are natural for humans: hue, saturation, and intensity [276].

One could argue, though, that the extended color vector approach cannot be merged with a human-centered approach: Since the statistical values that represent the color features are mathematical artifacts, they are not intuitive for humans. That this is not necessarily the case, is (at least to a certain degree) illustrated by the forthcoming dialog, which is a metaphor alluding the color matching process. Mr. Black and Mr. White are doing manual color based image retrieval:

White: *Hey, what about this picture? This picture has the same amount of green as the query image.*
Black: *That's right, but on average this green is a little bit more blue-ish than the image we are looking for.*
White: *I'm afraid, you're right. Also the red parts are in general more pale, I guess.*
Black: *True, and moreover these orange parts have a range of different tints. They do not appear in the original image.*
White: *Hmm, indeed... I think the image we had just before, fits the query image better, doesn't it?*
Black: *I totally agree. Let's put it under that one, in our retrieval stack.*

This dialog underlines that the natural components of the HSI color space [276] combined with the statistical measures, match human capabilities with respect to describing color features.

For optimal use of these extended features, we developed a new similarity measure that combines multiple color features per color category. This way of similarity matching is a top-down approach when the level of color feature abstraction is regarded. Let us discuss Figure 8.1 (adopted from Gong [94]) to clarify this.

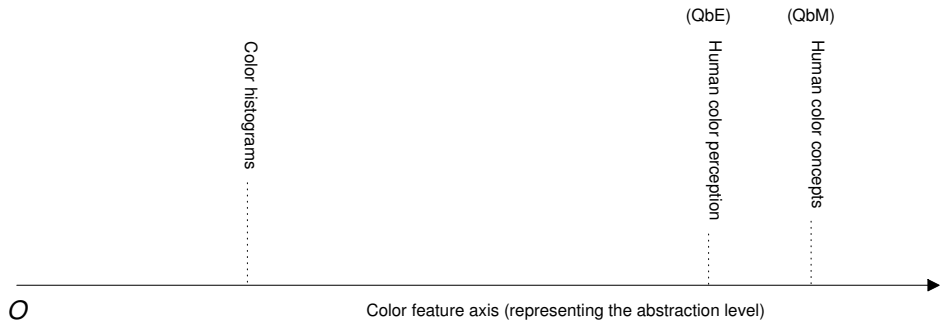


Figure 8.1: An axis denoting the abstraction level of color features (adopted from Gong [94]). Between parenthesis we added the query paradigms at the color abstraction level that they require.

Gong [94] describes a color feature axis in which he discerns levels of color feature abstraction. Three levels are made explicit: (i) color histograms, (ii) human color perception, and (iii) human color concepts (i.e., categories). Level (i) refers to histogram based CBIR techniques founded on arbitrary color space segmentation [287]. Human perception (level (ii)) and human color concepts (level (iii)) are related to two query paradigms that are used in CBIR, respectively query-by-example and query-by-memory.

Generally, CBIR research is approached bottom-up; many efforts are done to reach abstraction level (ii). Computational methods should reach that abstraction level, to ensure optimal CBIR performance in a query-by-example environment. PicHunter [63] and VisualSEEk [275] try to bridge the gap between level (i) and (ii) by using quantization methods that are more close to human perception. Nevertheless, these approaches do not close the color abstraction gap completely. More complex phenomena of human color perception, like the invariance for changes in image illumination, are not represented by them. As Mandal et al. [175] stated, "current histogram techniques retrieve similar images acquired under similar illumination levels.

As described in Chapter 5, we used human color categorization data to segment color space. By using this segmented color space for CBIR purposes, we can retrieve images based on color concepts. In other words, by segmenting color space into categories, we reached an abstraction level that comes close to the level of color concepts. As a result, we are perfectly able to fulfill the demands of the query-by-memory (QbM) paradigm by using histograms that represent color categories. In that sense, we closed the (abstraction) gap between color histograms and color concepts; see also Figure 8.1.

Next, we have expanded the CBIR query domain of our approach to the query-by-example (QbE) paradigm by using a top-down strategy; i.e., from level (iii) to (ii) in Figure 8.1. Note that this is an unconventional way to address the abstraction level of human color perception. However, since the color concepts influence the human color percep-

tion [318], correspondingly CBIR might take advantage from an analog top-down effect. In this top-down approach, we could exploit the advantage of the coarse segmentation of color space: The CBIR engine utilizes color categories, each covering a broad variety of color values. As a result, the image retrieval performance is robust with respect to various complex perceptual issues (e.g., caused by changes in image illumination). Consequently, modeling of these features is of less importance in the proposed top down approach.

In contrast to QbM where humans have to rely on their color memory, in a QbE environment they can directly differentiate color values of query and target images (see Chapter 5). So, the QbE paradigm requires color descriptors that reveal subtle color differences between images. As discussed earlier, we added mean and standard deviation values of the natural HSI color space, for each color category (also see Chapter 6). Matching of the mean values indicates the distance of the average color value between images per color category. Additionally, matching of the standard deviation of color values allows us to compare the coherence of color values (per category) between images. Because of the natural character of the HSI color space, these values can be perceived and discussed by humans, as was illustrated in the dialog in the beginning of this section. The additional color features supply color information that is much more subtle than solely the quantitative information of color categories. Hence, a downgrade effect toward the abstraction level of human color perception is established.

As discussed above, we added values of the HSI color space, which represents axes that are natural for humans. In the current implementation the impact on the similarity calculation is equal for each value; they all have the same weight. Nevertheless, the values for hue could get a greater magnitude since "hue represents the most significant characteristic of the color." [275] Possibly, this downgrade effect could be made even more strong by changing the weight of some of these additional features.

As measuring illumination can be a pitfall with regard to CBIR for QbE purposes in a bottom up approach, we avoided it by the exclusion of the mean intensity value (for each color category). Hence, up to a high extent our CBIR approach remains insensitive for changes in image illumination.

8.4 The CBIR benchmark

For the comparison of the three CBIR engines, a slightly modified version of the benchmark presented in Section 11.6 of Chapter 7 was used. Again the Corel image database, consisting of 60,000 images, served as our data-set. The engines to be tested are defined by two parameters: the color histogram of choice and the distance (or similarity) measure chosen. We will now discuss the three CBIR engines tested, using these two parameters.

8.4.1 Introduction: The 3 CBIR systems

For the first parameter, the color histogram database, two histogram configurations were used (11 and 4096 bins), each having their own quantization method. For the histogram configuration using 11 bins a quantization method was used based on the proposed segmented HSI color space. The second histogram configuration is the QBIC configuration using 4096 (16x16x16) bins [85, 193] determined in RGB color space. This computationally heavy configuration is chosen because it performed best in the benchmark described in the previous chapter.

For the second parameter, two histogram matching functions were used in our benchmark: (i) the histogram intersection distance [287] (see Equation 8.1) and (ii) the proposed similarity function, which combines intersection similarity (see Equation 8.2) and statistical information (see Section 8.2). We have used the intersection distance measure because it was judged as performing better than the quadratic distance for all histogram configurations [30].

The proposed similarity function was only applied on the 11 bin configuration. So, in total three engines (i.e., combinations of color quantization schemes and distance measures) are compared in the benchmark: i) the 4096 bin configuration, ii) the 11 bin configuration, iii) the enhanced 11 bin configuration, using the similarity function.

Table 8.1: The three engines incorporated in the CBIR-benchmark, defined by their color quantization scheme (i.e., 11 or 4096 bins) and the distance measure applied; i.e., the intersection distance (ID) or the extended (or enhanced) intersection distance (ID_e).

	11 bins	4096 bins
Intersection distance	$ID-11$	ID-4096
Extended intersection distance	ID_e-11	

8.4.2 Method

For each of the three engines, 30 query results had to be judged by human subjects, making a total of 90 per participant. They were unaware of the fact that three distinct engines were used to retrieve the images. Each set of 90 queries was fully randomized, to control for influence of order. Normally such retrieval results are presented in their ranked order. However, if this would have been the case in the benchmark, the participants would be biased to the first retrieval results after a few queries. Therefore, the ranking of the retrieved images is presented in random order.

Each query resulted in 15 retrieved images, presented in a 5×3 matrix. On the left side of this matrix the query image was shown (see Figure 8.2).

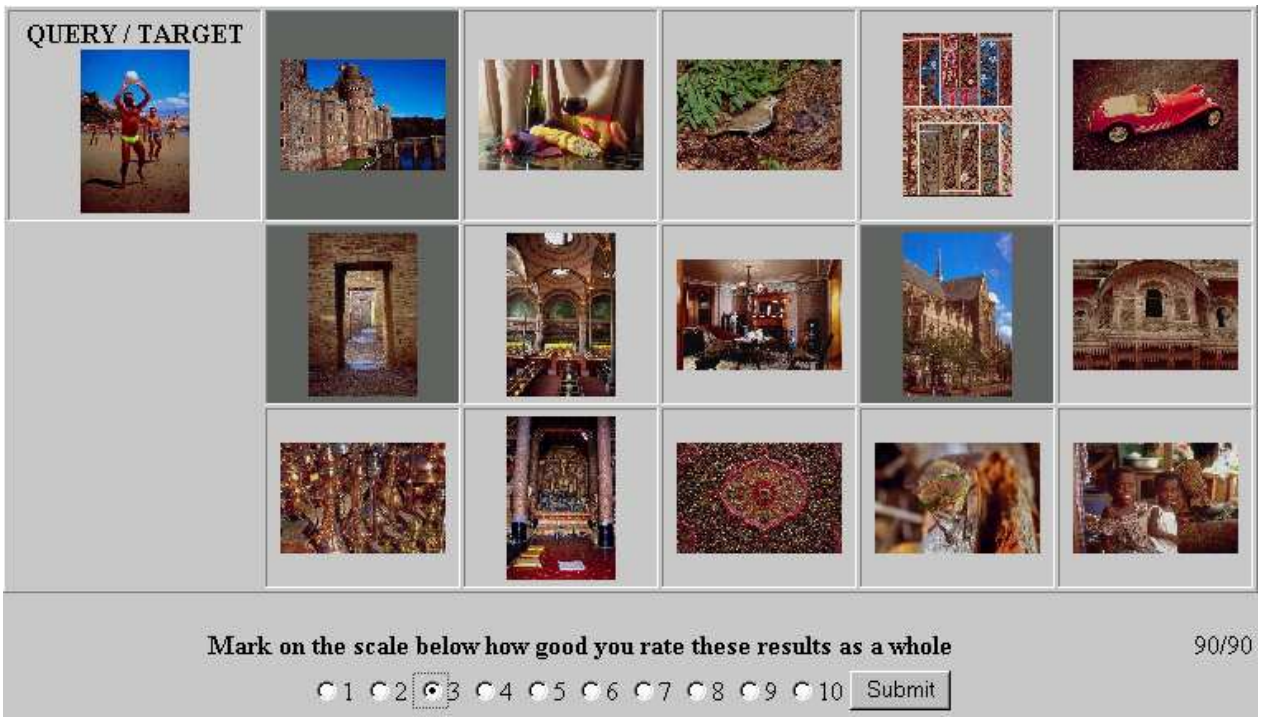


Figure 8.2: The interface of a query as was presented to the participants. They were asked to select the best matching images and to rate their overall satisfaction, with respect to their color distribution only. See Figure B.7 in Appendix B for a large full color print of this screendump.

The participants were asked to judge the retrieved images solely based on the color distribution hereby ignoring the spatial distribution of the colors. It was emphasized that semantics, shape, etc. should not influence their judgment. The participants were asked to perform two tasks. On the one hand they were asked to mark the images that they judged as relevant. On the other hand, they were asked to indicate their overall satisfaction with the retrieved results on a scale from 1 to 10 (see Figure 8.2). The complete instructions can be found on `http://eidetic.ai.ru.nl/CBIR-benchmark.html`.

51 participants, both males and females in the age of 20-60, finished the benchmark; 11 did start with the benchmark but did not finish it. The data of this second group of participants was not taken into account for analysis.

Regrettably, six of the 51 participants did not complete the benchmark as instructed. Five of these six participants did not select any of the retrieved images. One of the six participants consistently selected one image for each of the 90 query results. Therefore, these six participants were not taken into account for the analysis. Hence, in total we did collect usable data of 45 participants, making a total of 4050 queries that were judged.

We recorded for each query of each participant: the image ID, the query number, the number of bins used, whether or not the within bin statistics were used, the selected satisfaction rate, and which and how many images the participant judged as relevant.

Both the number of selected images and the rating for each query were normalized per participant. The normalized values were used for the analysis. How this normalization was done is defined in the Section 7.2.5 of the previous chapter.

8.4.3 Results

Two dependent variables resulted from the experiments: the number of images selected by the participants as acceptable and the overall rating given by the participants. Both measures were analyzed. The aim of the first phase of analysis was to determine whether a difference was present between the three engines. Therefore, for each measure a one-way ANOVA was applied. A strong and highly significant general difference was found between the engines for both the number of selected images ($F(2,4047)=60.29$; $p<.001$) and for the overall rating provided by the participants ($F(2,4047)=97.60$; $p<.001$).

A Duncan post hoc test on the number of selected images, revealed two homogeneous subsets within the group of three engines. According to the number of selected images, the $ID-11$ and ID_e-11 engines did not differ. Yet, this finding was not confirmed by the values on overall rating of the retrieval performance. A complete description of the statistical data can be found in Table 8.2.

Based on the latter result we conducted an additional analysis. Six additional ANOVAs were applied: each of the three engines was compared with the two others for both measures. According to the overall rating the within bin statistics had improved the performance of the $ID-11$ engine ($F(1,2698)=15.15$; $p<.001$). In contrast, on the number of selected images a non-significant ($F(1,2698)=3.00$; $p<.084$) was found. The complete results of the six ANOVAs can be found in Table 8.3.

Further, we were interested in the variability between participants. To determine a general effect of variability between participants, we applied a Multivariate ANOVA, which revealed, for both measures, a strong variability between participants (number of selected images: $F(1,4046)=10.23$; $p<.001$ and rating: $F(1,4046)=6.61$; $p<.010$).

Table 8.2: Descriptive statistics of the benchmark. Each engine is defined by its color quantization scheme ($\#bins$) and whether or not statistical data on bin level ($stats.$) was taken into account. In addition, the number of queries ($\#queries$) performed by each engine is mentioned. For the number of selected images (#images selected) as well as for the overall rating the mean (μ) value, the standard deviation (σ), and the confidence interval $(\min, \max)$ at 99% is provided, for each engine.

engine	#queries	μ	σ	$(\min, \max)$	μ	σ	$(\min, \max)$
$ID-11$	1350	3.67	3.51	3.42–3.93	4.76	2.29	4.60–4.92
ID_e-11	1350	3.91	3.48	3.65–4.17	5.10	2.28	4.94–5.26
$ID-4096$	1350	5.13	4.06	4.87–5.39	5.96	2.36	5.80–6.12

Table 8.3: Strength and significance of the difference found between the 11 bin, the enhanced 11 bin (including within bin statistical information: + stats.), and the 4096 bin engine, on both the number of selected images and the overall rating.

engine 1	engine 2	#images selected	rating
$ID-11$	ID_e-11	F(1,2698)= 3.00 (p<.084)	F(1,2698)= 15.15 (p<.001)
$ID-11$	$ID-4096$	F(1,2698)= 99.02 (p<.001)	F(1,2698)=181.23 (p<.001)
ID_e-11	$ID-4096$	F(1,2698)= 70.27 (p<.001)	F(1,2698)= 93.44 (p<.001)

Table 8.4: Strength and significance of the variability between participants for the 11 bin, the enhanced 11 bin (including within bin statistical information: + stats.), and the 4096 bin engine, on both the number of selected images and the overall rating.

engine	#images selected	rating
$ID-11$	F(1,1348)= 7.00 (p<.008)	F(1,1348)= 3.31 (p<.069)
ID_e-11	F(1,1348)= 5.83 (p<.016)	F(1,1348)= 2.42 (p<.120)
$ID-4096$	F(1,1348)= 0.47 (p<.493)	F(1,1348)= 1.19 (p<.276)

Three Multivariate ANOVAs were done to determine for each of the three engines, how much participants differ in their scores. A complete overview of the variability between the participants for each of the three engines is provided in Table 8.4.

8.4.4 Discussion

A large amount of data was collected through the benchmark, which is permanently available for participants at http://eidetic.ai.ru.nl/CBIR-benchmark.html. The results of this research comprise 4050 queries that were judged by 45 participants. Two measures were used: the number of selected images and the overall rating, indicating the satisfaction of the participant.

Without being told, the participants judged three distinct engines. Each engine can be defined by a combination of a color quantization measure and a distance measure. Two engines used the 11 bin quantization of color space introduced in Chapter 5 and the third one used the 4096 bin quantization of color space, adopted from IBM's QBIC [102, 274]. The latter was judged as performing best in the previous chapter. The 4096 bin quantization and one of the 11 bin quantizations, which we call the $ID-11$ engine, employed the intersection distance measure. The other 11 bin quantization was equipped with a newly developed similarity function, based on within bin statistical information, which we therefore name the ID_e-11 engine.

The original feature vector, based on the 11 color categories, was originally developed for the query-by-memory paradigm, as discussed in Section 7.1.1 of Chapter 7. With respect to the query-by-example paradigm, we developed an extended color feature vector, which

contains statistical color values for each color category and has four major advantages:

1. A rich source of additional information is available; i.e., an additional 43 color features.
2. The human-centered color space segmentation is utilized.
3. The CBIR engine is still computationally inexpensive [136]; i.e., only 54 dimensions.
4. The additional information is based on color features that are intuitive for humans: hue, saturation, and intensity [95, 276].

However, one could argue that the extended color vector approach can not be merged with a human centered approach: Since the statistical values are mathematical artifacts, which are not intuitive for humans. This is not necessarily the case, as illustrated in Section 8.3: the natural components of the HSI color space combined with the statistical measures, match human capabilities with respect to describing color features. [146]

The enhancement/extension of the 11 bin color quantization scheme combined with the new similarity measure improved the performance significantly compared to the standard 11 of bin color quantization scheme combined with the intersection measure. However, the 4096 bin engine performed best, according to the participants.

The advantage of the standard 11 bin approach in combination with the new similarity measure, is its low computational complexity, where it outperforms the 4096 bin histogram by far. Taking in consideration that the latter is of extreme importance [136] for the usage of CBIR systems, the results were very promising. However, it should be noted that a strong variability between the participants was found for all three engines, with respect to the number of images they selected (see Table 8.4). In contrast, the overall rating did not show a significant variability between the participants for any of the engines. So, a strong discrepancy was present between both measures, with respect to the variability between participants.

The participants reported that judging whether a retrieved image should be considered as relevant, is a particularly difficult process. This was mainly due to the fact that they were asked to judge the images based solely on their color distribution and to ignore their semantic content. Therefore, we have strong doubts concerning the reliability of the number of selected images as a dependent variable. The overall rating should be considered as the only reliable variable. For a benchmark such as ours, the number of selected images should, therefore, not be included in future research nor in further analysis of the current research.

With this second benchmark, we end our evaluations directed to image retrieval based on global color distributions. In the next chapters, a new line of research is explored. It continues the work discussed in this and the previous chapters. Again, the color space segmentation, as introduced in Chapter 5 will be utilized. However, in this third line of research another important feature used by CBIR engines will be explored: Texture.

9

Texture representations

Abstract

In the first part of this chapter, the concept texture is introduced and the difficulty of defining texture is illustrated. Next, a short description of the two approaches to texture analysis is given, followed by the introduction of a new, efficient algorithm for the calculation of the co-occurrence matrix, a gray-scale texture analysis method. The co-occurrence matrix is extended to color with the introduction of the color correlogram. Last, some applications of texture analysis are described and the classifiers used in this thesis to perform texture classification are introduced. In the second part of this chapter, first, a gray-scale study is done to determine the optimal configuration for the co-occurrence matrix on a texture classification task. Second, the co-occurrence matrix and the color correlogram were both used for the analysis of colorful texture images. The use of color improved the classification performance. In addition was found that coarse quantization schemes are more successful than more precise quantization schemes.

This chapter is partly based on:

Broek, E. L. van den and Rikxoort, E. M. van (2004). Evaluation of color representation for texture analysis. In R. Verbrugge, L.R.B. Schomaker, and N. Taatgen (Eds.), *Proceedings of the Belgian Dutch Artificial Intelligence Conference (BNAIC) 2004*, p. 35-42. October 21-22, Groningen - The Netherlands.

Texture is an intuitive concept that describes properties like smoothness, coarseness, and regularity of a region [95]. Texture is an important element to human vision, it provides cues to scene depth and surface orientation.

In the next sections, Intensity-based texture will be described, which has been the topic of investigation for many years and has proven useful. For example, the black and white television proves the usability of Intensity-based texture: people are able to see 3D in a 2D black and white screen. So, it seems important to look at Intensity-based textures before looking at colorful textures because the techniques used by Intensity-based textures can probably be expanded to color-texture.

9.1 Texture defined

This section provides an overview of the definitions of texture that have been proposed in the literature over the years. It is adopted from the web page of MeasTex [272] and after that extended.

Even though texture is an accepted intuitive concept, a formal definition of texture seems elusive. In 1973, Haralick, Shanmugam, and Dinstein [106] noted (p611): *"texture has been extremely refractory to precise definition"*. Over the years, many researchers have expressed this sentiment: Cross and Jain [64] (p.25): *"There is no universally accepted definition for texture."*, Bovik, Clarke, and Geisler [22] (p.55): *"an exact definition of texture either as a surface property or as an image property has never been adequately formulated."*, and Jain and Karu [127] (p.195): *"Texture [eludes] a formal definition"*. Standard works confirm this. Gonzales and Woods [95] (p.665) state:*"No formal definition of texture exists, intuitively this descriptor provides measures of properties such as smoothness, coarseness and regularity."* and Ballard and Brown [8] write: *" The notion of texture admits no rigid description, but a dictionary definition of texture as ' something composed of closely interwoven elements' is fairly apt."* The latter statement is confirmed by Merriam-Webster dictionary [181] which provides five definitions of texture, of which four are applicable for us:

1. a: *"something composed of closely interwoven elements; specifically: a woven cloth"* b: *"the structure formed by the threads of a fabric"*
2. a: *"essential part: substance"* b: *"identifying quality: character"*
3. a: *"the disposition or manner of union of the particles of a body or substance"* b: *"the visual or tactile surface characteristics and appearance of something, the texture of an oil painting "*
4. a: *"basic scheme or structure"* b: *"overall structure"*

Despite the lack of a universally agreed definition, all researchers agree on two points. Firstly, there is significant variation in intensity levels between nearby pixels; that is, at the

limit of resolution, there is non-homogeneity. Secondly, texture is a homogeneous property at some spatial scale larger than the resolution of the image.

Some researchers describe texture in terms of human visual perception: that textures do not have uniform intensity, but are none-the-less perceived as homogeneous regions by a human observer. For example, Bovik, Clarke, and Geisler [22] (p.55) write: *"an image texture may be defined as a local arrangement of image irradiances projected from a surface patch of perceptually homogeneous irradiances"*. Also, Chaudhuri, Sarkar, and Kundu [52](p.233) write: *"Texture regions give different interpretations at different distances and at different degrees of visual attention. At a standard distance with normal attention, it gives the notion of macro-regularity that is characteristic of the particular texture. When viewed closely and attentively, homogeneous regions and edges, sometimes constituting texels, are noticeable."* However, a definition based on human acuity poses problems when used as the theoretical basis for a quantitative texture analysis algorithm. Faugeras and Pratt [83](p.323) note: *"The basic pattern and repetition frequency of a texture sample could be perceptually invisible, although quantitatively present."*

9.2 Texture Analysis

There are two widely used approaches to describe the texture of a region, these are statistical and structural. The statistical approach considers that the intensities are generated by a two-dimensional random field. The methods used are based on spatial frequencies and yield characterizations of textures as smooth, coarse, grainy, etcetera. Examples of statistical approaches to texture analysis are autocorrelation function, gray-level co-occurrence matrix, Fourier texture analysis, edge frequency, and Law's texture energy measures [257, 279].

The structural techniques deal with the arrangement of image primitives, such as the description of texture based on regularly spaced, parallel lines [279]. In our research, the co-occurrence matrix was used to perform texture analysis because it is an important gray-scale texture analysis method [95, 106]. As Palm [205] states: 'several studies favor the co-occurrence matrices in the gray-scale domain'. In addition, Sharma, Markou, and Singh [257] performed a comparative study using five texture analysis methods and found the co-occurrence matrix outperformed the other methods.

9.3 The co-occurrence matrix

The co-occurrence matrix is constructed from an image by estimating the pairwise statistics of pixel intensity. In order to (i) provide perceptual intuitive results and (ii) tackle the computational burden, intensity was quantized into an arbitrary number of clusters of intensity values, which we will name: gray values.

The co-occurrence matrix $C_{\bar{d}}(i,j)$ counts the co-occurrence of pixels with gray values i and j at a given distance $\bar{d}$. The distance $\bar{d}$ is defined in polar coordinates (d,α), with discrete length and orientation. In practice, α takes the values $0°, 45°, 90°, 135°, 180°, 225°, 270°, \text{ and } 315°$. The co-occurrence matrix $C_{\bar{d}}(i,j)$ can now be defined as follows:

$$C_{\bar{d}}(i,j) = \Pr(\mathrm{I}(p_1) = i \wedge \mathrm{I}(p_2) = j \mid |p_1 - p_2| = \bar{d}), \tag{9.1}$$

where $\Pr$ is probability, and p_1 and p_2 are positions in the gray-scale image I.

The algorithm yields a symmetric matrix, which has the advantage that only angles up to $180°$ need to be considered. A single co-occurrence matrix can be defined for each distance ($\bar{d}$) by averaging four co-occurrence matrices of different angles (i.e., $0°$, $45°$, $90°$, and $135°$).

Let N be the number of gray-values in the image, then the dimension of the co-occurrence matrix $C_{\bar{d}}(i,j)$ will be $N \times N$. So, the computational complexity of the co-occurrence matrix depends quadratically on the number of gray-scales used for quantization.

Because of the high dimensionality of the matrix, the individual elements of the co-occurrence matrix are rarely used directly for texture analysis. Instead, a large number of textural features can be derived from the matrix, such as: energy, entropy, correlation, inverse difference moment, inertia, Haralick's correlation [106], cluster shade, and cluster prominence [60]. These features characterize the content of the image.

Note that in order to apply the co-occurrence matrix on color images, these images have to be converted to gray-value images. This conversion is described in Section 2.3, for several color spaces.

9.4 Colorful texture

For gray-scale image analysis multiple are algorithms available, for color images this is not yet the case. However, new algorithms for color texture analysis are increasingly explored as color analysis becomes feasible [267]. One of the approaches that has been proposed several times is the so called color correlogram [118, 119, 267]. The color correlogram is an extension of the co-occurrence matrix to color images and was first proposed by Huang et al. [118].

Huang et al. [118, 119] used the color correlogram as a feature for image indexing and supervised learning. They found that the color correlogram performed better than well known methods like the color histogram. Therefore, in this research the color correlogram is used as a color texture descriptor.

The color correlogram is the colorful equivalent of the co-occurrence matrix. The color correlogram $C_{\bar{d}}(i,j)$ counts the co-occurrence of colors i and j at a given distance $\bar{d}$. So,

for the color correlogram, not the intensity is quantized, but a color space is quantized. In Equation 9.1, i and j denote two gray-values. Subsequently, the color correlogram can be defined by Equation 9.1, with i and j being two color values.

9.5 Using texture

Texture analysis can be used for many applications. There are six categories of utilizations of texture analysis, these are texture segmentation, texture classification, texture synthesis, 2D and 3D shape extraction from texture, and motion-from-texture [299]. A brief description of these different utilizations is given below.

1. **Texture segmentation** is the problem of trying to find different textures in one image. This is a difficult problem because usually one does not know how many different textures there are in an image, and what kind of textures, etc. But this knowledge is not necessary if there is a way to tell that two textures are different.
2. **Texture classification** involves deciding in which texture category a given texture belongs. In order to do this, the different texture categories need to be known; e.g., rock, grass, fabric, clouds. Pattern recognition classifiers, like neural networks or Bayesian classifiers, can be used to classify the textures.
3. **Texture synthesis** is the problem of synthesizing a new texture, from a given texture, that, when perceived by a human observer, appears to be generated by the same underlying stochastic process. Texture synthesis can, for example, be used for image de-noising and image compression.
4. **2D shape extraction from texture** can be applied next to texture segmentation. Based on local differences in color/intensity and texture shapes can be extracted from images.
5. **3D shape extraction from texture.** Various visual cues are used by humans to recover 3D information from 2D images. One such cue is the distortion of textures due to the projection of the 3D world onto a 2D image plane. This distortion can be used to recover shape from texture.
6. **Motion-from-texture** is the least frequently used application of texture. However, Werkhoven, Sperling, and Chubb [314] showed that drifting spatiotemporal modulations of texture can induce vivid motion percepts.

In this study, texture will be used for the first three utilizations: segmentation, shape extraction, and classification.

9.6 Three classifiers

The classification of texture can be done with different classifiers. Based on initial pilot experiments, in which different classifiers were compared, three classifiers were chosen:

1. **The statistic classifier** is based on discriminant analysis with linear discriminant function y_k that decides on class membership. An input vector x is assigned to a class C_k if $y_k(x) > y_j(x)$, for all $j \neq k$.
2. **The probabilistic neural network** approximates the probability density function of the training examples presented. It consists of three layers after the input layer: the pattern layer, the summation layer, and the output layer. The outcome is a classification decision in binary form.
3. **The K-nearest neighbor classifier** works with the following algorithm: suppose the data set contains N_k data points in class C_k and N points in total, so that $\sum_k N_k = N$. The classifier then works by drawing a hypersphere around the point to classify, x, which encompasses K points. To minimize the probability of misclassifying x, x is assigned to the class C_k for which the ratio $\frac{K_k}{K}$ is largest, where K_k is the number of points from class C_k.

They were combined using the technique of majority voting [147]: when at least two of the three classifiers agree on the class label of a sample texture, this label is given else the label reject is given. An explanation for the choice to use them separately or to combine them is provided in the chapters where the classifiers are applied. The same holds for the parameter settings.

9.7 The exploratory gray-level study

A pilot study was done to explore the possibilities of the gray-level co-occurrence matrix as described in Section 9.3. In this study, a classification task on the MeasTex gray-scale texture database [272] is performed to determine the optimal set of textural features to be derived from the co-occurrence matrix (see Section 9.3). This is needed since no consensus is present in literature on which feature (combination) describes texture best. In addition, an attempt is made to describe what these features represent for humans.

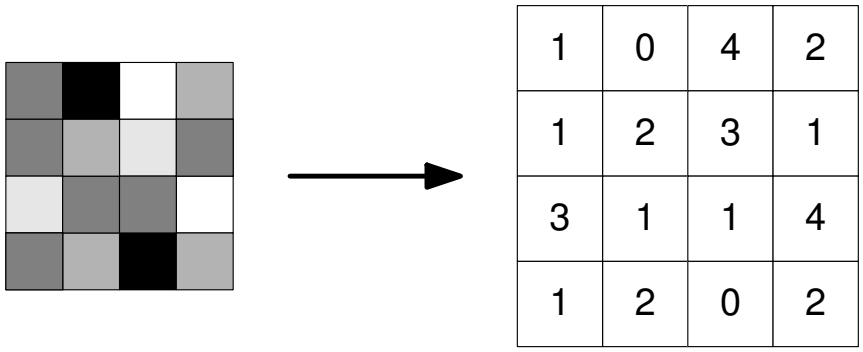
9.7.1 The features

As described in Section 9.3, features can be extracted from the co-occurrence matrix to reduce feature space dimensionality and so, reduce the computational complexity. Figures 9.1a–9.1d visualize the relation between the features energy and entropy and between inertia (or contrast) and Inverse Difference Moment (IDM). In the next section, a textual description is provided of each of these features. This section provides the formal definitions of eight features from the co-occurrence matrix.

$$
\begin{aligned}
\text{Energy} &= \sum_{i,j} C(i,j)^2 \\
\text{Entropy} &= -\sum_{i,j} C(i,j) log C(i,j) \\
\text{Inverse Difference Moment} &= \sum_{i,j} \tfrac{1}{1+(i-j)^2} C(i,j) \\
\text{Inertia (or contrast)} &= \sum_{i,j} (i-j)^2 C(i,j) \\
\text{Cluster Shade} &= \sum_{i,j} ((i-\mu_i)+(j-\mu_j))^3 C(i,j) \\
\text{Cluster Prominence} &= \sum_{i,j} ((i-\mu_i)+(j-\mu_j))^4 C(i,j) \\
\text{Correlation} &= \sum_{i,j} \tfrac{(i-\mu_i)(j-\mu_j)C(i,j)}{\sigma_i \sigma_j} \\
\text{Haralick's correlation} &= \tfrac{\sum_{i,j}(ij)C(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}
\end{aligned}
\tag{9.2}
$$

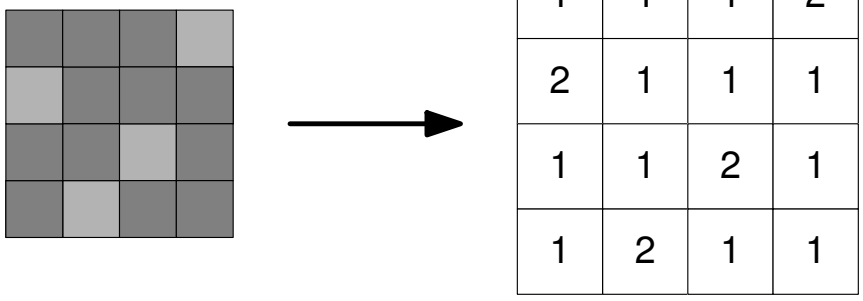
Notation

$C(i,j)$	the $(i,j)th$ entry in a co-occurrence matrix C
$\sum_i$	defined as: $\sum_{i=1}^{i=M}$ where M is the number of rows.
$\sum_j$	defined as: $\sum_{j=1}^{j=N}$ where N is the number of columns.
$\sum_{i,j}$	means $\sum_i \sum_j$
μ_i	defined as: $\mu_i = \sum_i i \sum_j C(i,j)$
μ_j	defined as: $\mu_j = \sum_j j \sum_i C(i,j)$
σ_i	defined as: $\sigma_i = \sum_i (i-\mu_i)^2 \sum_j C(i,j)$
σ_j	defined as: $\sigma_j = \sum_j (j-\mu_j)^2 \sum_i C(i,j)$
μ_x, μ_y	the mean of row and column sums respectively.
σ_x, σ_y	the standard deviation of row and column sums respectively.



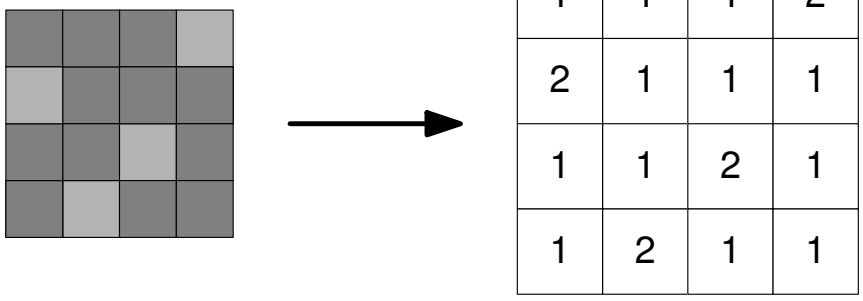
1	0	4	2
1	2	3	1
3	1	1	4
1	2	0	2

(a)



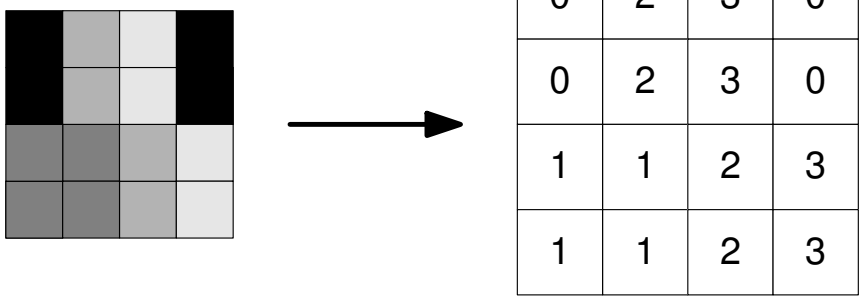
1	1	1	2
2	1	1	1
1	1	2	1
1	2	1	1

(b)



1	1	1	2
2	1	1	1
1	1	2	1
1	2	1	1

(c)



0	2	3	0
0	2	3	0
1	1	2	3
1	1	2	3

(d)

Figure 9.1: The relation between energy an7 entropy: (a) energy low and entropy high and (b) energy high and entropy low. The relation between inertia (or contrast) and Inverse Difference Moment (IDM): (c) inertia low and IDM high and (d) inertia high and IDM low.

9.7.2 The features describing texture

In the latter section, the formal definitions of eight features were given. In the current section, the intuitive descriptions of these definitions, are given.

Energy: describes the uniformity of the texture. In a homogeneous image, there are very few dominant gray-tone transitions, hence the co-occurrence matrix of this image will have fewer entries of large magnitude. So, the energy of an image is high when the image is homogeneous.

Entropy: measures the randomness of the elements in the matrix, when all elements of the matrix are maximally random, entropy has its highest value. So, a homogeneous image has a lower entropy than an inhomogeneous image. In fact, when energy gets higher, entropy should get lower.

Inverse difference moment: has a relatively high value when the high values of the matrix are near the main diagonal. This is because the squared difference $(i-j)^2$ is smaller near the main diagonal, which increases the value of $\frac{1}{1+(i-j)^2}$.

Inertia (or contrast): gives the opposite effect compared to the inverse difference moment, when the high values of the matrix are further away from the main diagonal, the value of inertia (or contrast) becomes higher. So, inertia and the inverse difference moment are measures for the distribution of gray-scales in the image. Also, when inertia is high, there will be small regions in the texture with the same gray-scale, for inverse difference moment, the opposite is true.

Cluster shade and Cluster prominence: are measures of the skewness of the matrix, in other words the lack of symmetry. When cluster shade and cluster prominence are high, the image is not symmetric. In addition, when cluster prominence is low, there is a peak in the co-occurrence matrix around the mean values, for the image this means that there is little variation in gray-scales.

Correlation: measures the correlation between the elements of the matrix. When correlation is high the image will be more complex than when correlation is low.

Haralick's correlation: is a measure of gray level linear dependence between the pixels at the specified positions relative to each other. Compared to normal correlation, Haralick's correlation reacts stronger to the complexity of an image.

9.7.3 The implementation of the co-occurrence matrix

A new, efficient algorithm for calculating the co-occurrence matrix is developed. A straightforward way to calculate the co-occurrence matrix would be to consider each pixel p_i of gray-level i and count all p_j of gray-level j at an arbitrary distance $\bar{d} = (d, \alpha)$, with $|p_i - p_j| = \bar{d}$,

for all possible i and j. This would take $O(nmb[\bar{d}])$ to compute, where $n \times m$ denote the image size, b is the number of gray levels distinguished, and $[\bar{d}]$ is the number of distances used, in our case four.

Our algorithm makes use of the definition of the co-occurrence matrix, instead of considering each gray-scale sequentially; it counts the co-occurrence of pixels with gray-values i and j at distance $\bar{d}$. The image is only scanned once, where the straightforward algorithm goes through the image $b[\bar{d}]$ times. For each pixel in the image, the pixels at the four distances $\bar{d}$ are considered. This is directly stored in the co-occurrence matrix in one iteration.

Let I denote a gray-scale image and $(\delta i_\alpha, \delta j_\alpha)$ be the displacement vector in the image (I) to obtain the pixel at distance $\bar{d}$ = $(1, \alpha)$ from pixel $I[i][j]$. Let C be the co-occurrence matrix for all four angels (i.e., $0°$, $45°$, $90°$, and $135°$) as described in Equation 9.1, which is initialized by setting all entrances to zero. Then, our algorithm is as follows:

for (i = 0; i< image_width; i++)
  for(j=0; j< image_height; j++)
    foreach $\alpha \in \{0°, 45°, 90°, \text{and } 135°\}$
      $C\ [I[i][j]]\ [I[i + \delta i_\alpha\ ][j + \delta j_\alpha\ ]]\ \ += 1;$

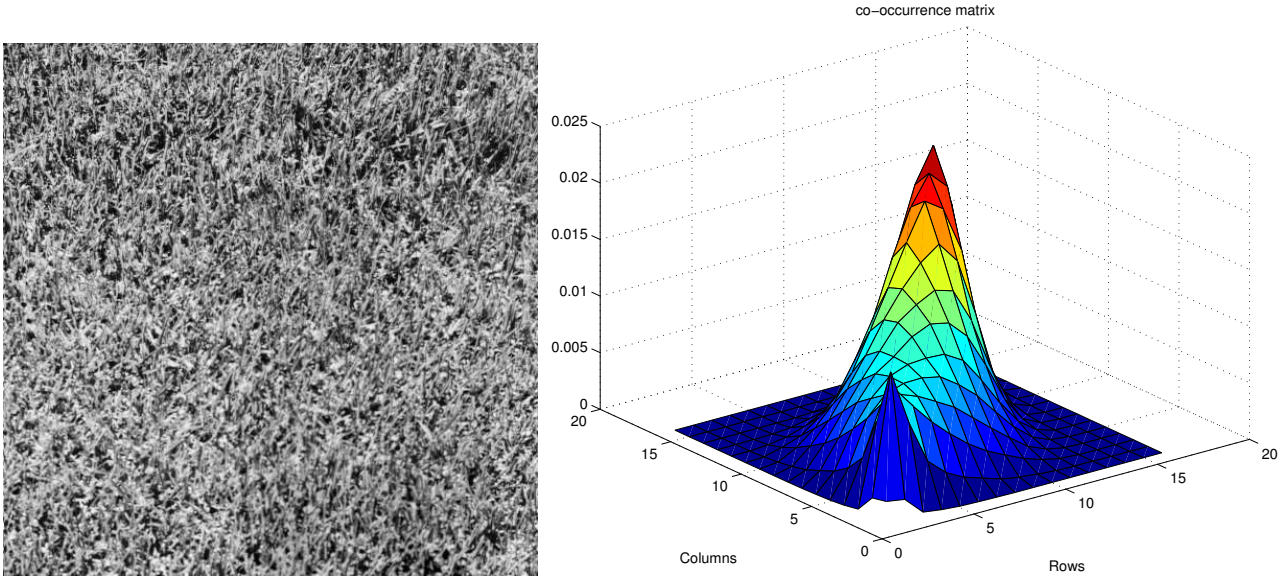
This algorithm takes $O(nm)$ time to compute.

9.7.4 Design

The co-occurrence matrix and the eight features were implemented as defined in Equation 9.1 and Equation 9.2. The intensity values were quantized in 64 bins using the RGB color space. The co-occurrence matrix was only calculated with distance 1, this is a common thing to do [106, 118]. As a database, the MeasTex gray-scale texture database [272] was used.

The MeasTex texture database is divided into five groups: asphalt, concrete, grass, misc, and rock. For this pilot study, only the groups concrete (12 images), grass (18 images), and rock (25 images) are used because the groups asphalt (4 images) and misc (9 images) are too small. The images have a size of 512×512 pixels.

In order to determine the optimal set of features for the co-occurrence matrix, a classification task was performed using a statistic classifier, as described in Section 9.6. Classification was applied using different combinations of the features. The classifier used is the discriminant analysis classifier from the statistic toolbox in Matlab, this classifier is used because it is thoroughly tested and has a good reputation.



(a) (b)

Figure 9.2: (a) A texture image from which a co-occurrence matrix and features are calculated. (b) The visual rendering of the co-occurrence matrix of image (a). See Figure B.5 in Appendix B for color versions of these images.

9.7.5 Results

To illustrate the texture analysis, in Figure 9.2 a visual rendering of the co-occurrence matrix of an image in the MeasTex database is given. In Table 9.1, the features for this image are given.

The classification task was performed by the linear discriminant analysis classifier, using all 255 (i.e., $2^8 - 1$ combinations) possible combinations of features. Four of the eight features used are found to be the strongest describers of texture. These are entropy, inverse difference moment, cluster prominence, and Haralick's correlation. When these features were used, 100% of the data was classified correct. When all the features were used, only

Table 9.1: The textural features derived from the image in Figure 9.2.

Textural features	Angle				
	$0°$	$45°$	$90°$	$135°$	$average$
Inertia (or contrast)	6.03	7.89	4.41	9.17	6.88
Energy	0.01	0.01	0.01	0.01	0.01
Entropy	4.71	4.83	4.58	4.88	4.75
Inverse Difference Moment	0.40	0.38	0.48	0.35	0.40
Cluster Shade	0.77	0.77	0.77	0.76	0.77
Cluster Prominence	0.70	0.70	0.70	0.70	0.70
Correlation	0.83	0.83	0.83	0.83	0.83
Haralick's Correlation	0.72	0.64	0.80	0.58	0.68

66% of the data classified correct. There is no risk of overfitting because a linear discriminant function was used to classify the data. A complete description of the results can be found in Van Rikxoort and Van den Broek [223].

9.7.6 Conclusion

The entropy, inverse difference moment, cluster prominence, and Haralick's correlation were found to be good descriptors of gray-scale texture. In addition, an efficient way of implementing the co-occurrence matrix is presented. Now, it is interesting to look at colorful textures, using the color correlogram.

Huang et al. [118, 119] used the color correlogram in a whole as a feature. Like the co-occurrence matrix, this is a very high dimensional feature. Hence, it is desirable to reduce the dimension. Extracting features from the color correlogram is a method of reducing this dimension [205]. The features used for the gray-level co-occurrence matrix can be expanded to deal with the color correlogram [267]. Alternatively, color images can be read and converted to gray-scale images before calculating the features. This conversion can, using the RGB color space, be made by the following formula [89]:

$$I = \frac{R + G + B}{3}, \tag{9.3}$$

where I denotes Intensity or gray-value. The conversion from color to gray-scale for other color spaces is given in Section 2.3.

When analyzing colorful texture, the color space used will be quantized. However, it is not known how many different colors exactly influence the appearance of a texture to humans. There has been some research that suggests that only a few different colors are important to human color perception [16, 75, 219].

Finally, it seems a good idea to test the effect of using different color spaces for representing colorful texture. Sing et al. [267] used 11 different color spaces and found that all gave good results for texture analysis. However, there were very different classification performances observed, using the same texture features and the same experimental data, depending on which color space was used.

9.8 Exploring colorful texture

In this study, the color correlogram and co-occurrence matrix are used for the same colorful texture classification task. Hence, the performance of the color-correlogram is compared to the co-occurrence matrix. Huang et al. [119] quantized the image in 64 different colors using the RGB color space. In the current research, a number of different quantizations were used.

They were initially compared using the RGB color space. In the next chapter, other color spaces are considered too.

9.8.1 Design

To implement the color correlogram, the efficient implementation of the co-occurrence matrix (Section 9.7.3) was used. Huang et al. [118] propose another algorithm to implement the color correlogram efficiently. Their algorithm also takes $O(mn)$ (where $m \times n$ are the image geometries) time to compute. Although the algorithm proposed by Huang et al. is equally efficient to the algorithm as introduced in Section 9.7.3, the latter is preferred because it is more intuitive.

In this study, the performance of the co-occurrence matrix on colorful textures is compared to the performance of the color correlogram. In order to apply the co-occurrence matrix on the colorful textures, the textures are converted to gray-scale using the RGB color space, as described in Section 9.7.6.

For both the co-occurrence matrix and the color correlogram, the four features performing best in the gray-scale pilot study are extracted. As a database, the VisTex colorful texture database [178] is used. The VisTex database consists of 19 labeled classes, the classes that contain less than 10 images were not used in this study, which results in four classes: bark (13 images), food (12 images), fabric (20 images), and leaves (17 images).

A classification task is performed, using the three classifiers described in Section 9.6. The training and test-set for the classifiers are determined using random picking, with the prerequisite that each class had an equal amount of training data. As a statistic classifier, the statistic classifier from the Matlab "statistic toolbox" is used. The Probabilistic neural network used in this study is implemented in the "neural network toolbox" from Matlab. The K-nearest neighbor classifier is obtained from the Matlab Central File Exchange [292]. There is an optimum for the value of K specific for each dataset. For this study, a value of $K = 2$ is chosen because of the small number of images.

9.8.2 Results

In Table 9.2 and 9.3, for both the co-occurrence matrix and the color correlogram, the percentages of correct classification are given for each of the three classifiers and for their combination. The classifiers were combined using the technique of majority voting, as described in Section 9.6. Combining the classifiers provides better results than using either of the classifiers separately. The results indicate that the quantization scheme used is important for texture analysis. The color correlogram performs better than the co-occurrence matrix when a coarse quantization is used.

Table 9.2: The percentages of correct classification for each classifier and their combination, for the co-occurrence matrix.

# Gray levels	Statistic	PNN	Knn	Combined
8	44%	66%	66%	66%
16	44%	66%	66%	66%
32	55%	55%	55%	55%
64	66%	44%	55%	55%
128	66%	55%	44%	66%

Table 9.3: The percentages of correct classification for each classifier and their combination, for the color correlogram.

# Levels of color	Statistic	PNN	Knn	Combined
8	55%	77%	55%	77%
64	55%	77%	77%	77%
256	55%	55%	55%	55%
512	55%	44%	44%	44%
1024	66%	33%	44%	44%

9.8.3 Discussion

The color correlogram was compared to the co-occurrence matrix, using several quantization schemes on the RGB color space. The color correlogram performed better than the co-occurrence matrix. This can be explained by the fact that no differentiation is possible between two colors with an equal intensity. Hence, the color of a texture is important for its classification, which is confirmed by Palm [205].

For both the color correlogram and the co-occurrence matrix, the best classification results are obtained using coarse quantization schemes. This is consistent with the concept of 11 color categories, as discussed in Chapters 2-8. The images in the VisTex image database are classified by humans, so it is plausible that they are classified using those color categories or at least into a limited number of color categories. In addition, coarse color quantizations are computationally cheap. This, in contrast with the quantization schemes proposed by Mäenpää and Pietikäinen [172], who used quantization schemes with up to 32768 color bins.

When the classifiers are combined, the results are better than when the three different classifiers are used separately. This agrees with the results found by Schiele that using multiple classifiers increases classification performance [243]. Therefore, in the remainder of this thesis, only the classification results of the classifier combination are given.

In the next chapter, we will continue our exploration of color-based texture analysis. Several color spaces, quantization schemes, and texture analysis techniques are explored. Moreover, a new method for combining color analysis and texture analysis is introduced.

10

Parallel-Sequential Texture Analysis

Abstract

In this chapter, color induced texture analysis is explored, using two texture analysis techniques: the co-occurrence matrix and the color correlogram as well as color histograms. Several quantization schemes for six color spaces and the human-based 11 color quantization scheme have been applied. The VisTex texture database was used as test bed. A new color induced texture analysis approach is introduced: the parallel-sequential approach; i.e., the color correlogram combined with the color histogram. This new approach was found to be highly successful (up to 96% correct classification). Moreover, the 11 color quantization scheme performed excellent (94% correct classification) and should, therefore, be incorporated for real-time image analysis. In general, the results emphasize the importance of the use of color for texture analysis and of color as global image feature. Moreover, it illustrates the complementary character of both features.

This chapter is almost identical to:

E. L. van den Broek and E. M. van Rikxoort. Parallel-Sequential Texture Analysis, *Lecture Notes in Computer Science (Advances in Pattern Recognition), 3687*, 532–541.

10.1 Introduction

There is more with colors than one would think at a first glance. The influence of color in our everyday life and the ease with which humans use color are in stark contrast with the complexity of the phenomenon color, a topic of research in numerous fields of science (e.g., physics, biology, psychology, and computer science). Despite their distinct views on color, scientists in these fields agree that color is of the utmost importance in image processing, both by humans and by computers. However, the use of color analysis increases the computational cost for image analysis algorithms, since instead of one dimension, three dimensions are present. Therefore, color images are often converted to gray-scale images, when texture analysis has to be performed (e.g., see Figure 10.2). Not surprisingly, with this conversion texture information is lost; e.g., using a standard conversion, red, green, and blue can result in the same gray-scale. Nevertheless, as Palm [205] already denoted: "The integration of color and texture is still exceptional". However, in the literature three distinct approaches to combine color and texture can be found: parallel, sequential, and integrative [205]. In the parallel approach, color and texture are evaluated separately, as shown in Figure 10.2. Sequential approaches use color analysis as a first step of the process chain: After the color space is quantized, gray-scale texture methods are applied, as shown in Figure 10.3. The integrative method uses the different color channels of an image and performs the texture analysis methods on each channel separately.

Palm [205] used an integrative method to test classification results on color textures and found that the use of color improved classification performance significantly. Drimbarean and Whelan [80] used three texture analysis methods on five different color spaces, with one (coarse) color quantization scheme in an integrative method to test classification results. The use of color improved performance, but no single color space outperformed the others. The results presented in the previous chapter confirm the additional value of color for texture analysis of color images. Mäenpää and Pietikäinen [172] used five different color spaces and two texture analysis techniques to determine whether color and texture should be used in parallel or sequential. They concluded that combining color and texture gave only minimal performance improvement, and that, when combining color and texture, the sequential approach should be preferred.

However, no reports are available that combine studies toward the influence of varying the color space, the quantization scheme, and the way color and texture are combined, for either the parallel approach, the sequential approach, or a combined approach. In this chapter, each of these variations is applied. Moreover, the new parallel-sequential approach is introduced: the color correlogram combined with the color histogram.

In the next section, the color spaces and the quantization schemes applied on them are described together with the color and texture analysis techniques (i.e., the co-occurrence

matrix, the color histogram, and the color correlogram), the texture database, and the classifiers used. As baselines, the co-occurrence matrix, the color histogram, and the color correlogram are applied, in Section 10.3. In Section 10.4, the new parallel-sequential approach is introduced and directly compared with the parallel approach. We end this chapter with a conclusion.

10.2 Method

Texture can be analyzed, using a simple color to gray-scale conversion or a color quantization scheme, as discussed in the previous chapter. Several texture analysis techniques have been developed, both for general and for specific purposes. One of the more intuitive texture descriptors is the co-occurrence matrix [106], which was developed for intensity based texture analysis. However, it can also be applied for colorful texture analysis; then it is denoted as the color correlogram [118], a sequential colorful texture analysis method: first, color is quantized and second, texture is analyzed. In Chapter 9, we determined which feature-distance combinations, derived from the co-occurrence matrix or color correlogram, perform best. The best classification was found for a combination of four features: entropy, inverse difference moment, cluster prominence, and Haralick's correlation, with $d = 1$. Subsequently, this configuration was also chosen for the current line of research.

For both the co-occurrence matrix and the color correlogram, the color spaces RGB, HSV, YIQ, YUV, XYZ, and LUV were used (see also Chapter 2). In addition, the color correlogram was combined with the human-based 11 color categories [16, 75, 219]. A complete overview of the schemes applied is presented in Table 10.1. In total, 170 different configurations were applied: 30 for the co-occurrence matrix, 20 for the color histogram, 45 for the color correlogram, and 75 for the combined approaches.

The VisTex texture database [178], which consists of 19 labeled classes, was used as test bed both for the baselines (see Section 10.3) and for the comparison between the parallel and

Table 10.1: The quantization schemes applied on the six color spaces and on the 11 color categories, for each texture descriptor. Note that YUV* is sampled for the color correlogram (see Section 2.3).

Color space	Co-occurrence matrix	Color histogram / Color correlogram
RGB	8, 16, 32, 64, 128	8, 64, 216, 512, 4096
HSV	8, 16, 32, 64, 128	27, 54, 108, 162, 324
YIQ, YUV*, XYZ, LUV	8, 16, 32, 64, 128	8, 27, 64, 125, 216
11 colors		11, 27, 36, 70, 225

parallel-sequential approach for texture analysis (see Section 10.4). The classes with less than 10 images were not used in this experiment. This resulted in four classes: bark (13 images), food (12 images), fabric (20 images), and leaves (17 images). In order to generate more data for the classifiers, we adapted the approach of Palm [205] and Mäenpää and Pietikäinen [172]: the original images were split into four sub-images, resulting in a database of 248 textures.

For all research described in this chapter, a combination of three classifiers, as described in Chapter 9, was used: a linear discriminant classifier, a 1-nearest neighbor classifier, and a probabilistic neural network, taken from the MATLAB® library using their default parameters. The output of this classifier combination was determined using the technique of majority voting [147]: when at least two of the three classifiers agree on the class label of a sample image, this label is given else the label false is given. The training and test set for the classifiers were composed using random picking, with the prerequisite that each class had an equal amount of training data.

10.3 Three baselines

As a first baseline, the co-occurrence matrix as standard, intensity-based texture analysis is used. The results are presented in Table 10.2. The complete results are available online [37]. The CIE LUV quantized in 8 bins and the HSV color space quantized in 32 bins performed best with a classification performance of 58%. Overall, the performances among different color spaces were about the same. Hence, for intensity-based texture analysis, the choice of color space is not crucial. The quantization scheme chosen is important, usually a lower number of bins performs better: In no instance, the largest number of bins gave the best results.

Table 10.2: The *best* classification results (%) of the color histogram, the co-occurrence matrix, and the color correlogram, for several color space - quantization scheme (#bins) combination.

Color space	Co-occurrence matrix		Color histogram		Color correlogram	
	#bins	%	#bins	%	#bins	%
RGB	8	56%	4096	87%	8	68%
HSV	32	58%	27	88%	162	74%
YIQ	8	54%			125	53%
YUV 4:4:4	8	54%			27	52%
XYZ	64	56%			27	71%
LUV	8	58%	64	84%	27	66%
11 colors			11	84%	27	72%

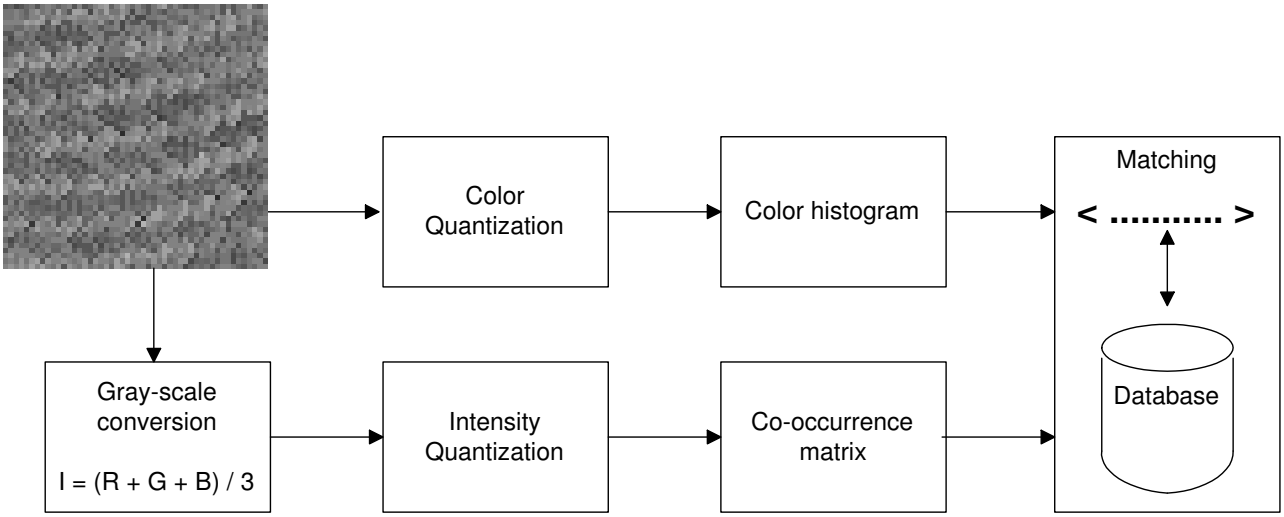


Figure 10.1: The parallel approach for texture analysis, using global color features and local intensity differences. In parallel, the color histogram is determined, after the quantization of color, and the co-occurrence matrix is calculated, after the conversion to gray-scale and the quantization of gray values.

Next to texture, the global color distribution within an image is frequently used as feature for classification and retrieval of images. Therefore, as a second baseline, we conducted an image classification experiment, using color solely by calculating the color histograms. In Table 10.2, the best four classification results are presented. The complete results are available online [37]. Classification by use of quantizations of the RGB color space results in a low performance (i.e., ranging from 19–48%), except for the 4096 bin quantization scheme (as used in QBIC [270]). However, the latter suffers from an unacceptable computational load, especially for real-time image analysis applications (e.g., content-based image retrieval). Therefore, the RGB color space is not suitable for color-based image classification. The classification using the coarsest LUV quantization (8 bins) had a poor performance. All other quantizations, using the LUV color space, resulted in high classification performance. The color-based texture classification, using the coarse 11 color quantization scheme, performed well (84%) (see Table 10.2), especially when considering its low computational complexity. The 27 bins quantizations of the HSV color space performed best with 88%.

As the third baseline, sequential texture analysis is performed (see Figure 10.3), with the color correlogram using six different color spaces. The results are presented in Table 10.2. In addition, the 11 color categories scheme was applied using several quantization schemes (see Section 13.4). The HSV color space performed best in combination with the color correlogram (see Table 10.2). This can be explained by the relatively high precision in color (Hue) quantization of the HSV 162 bins scheme. However, the color correlogram founded on the 11 color categories also performed good with 72% precision.

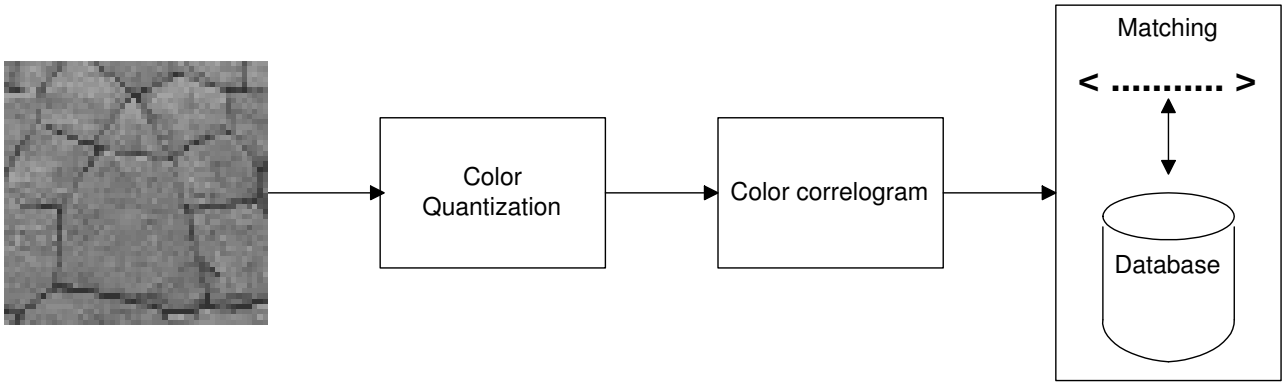


Figure 10.2: The sequential approach for texture analysis: after color quantization the color correlogram is utilized.

In line with the findings presented in the previous chapter, using more bins does usually not improve performance. In no instance, the largest number of bins gave the best results. This result emphasizes the importance of using a coarse color quantization scheme such as that of the 11 color categories in which one can represent colors (see also Chapter 2).

10.4 Parallel-sequential texture analysis: color histogram & color correlogram

In the previous sections, we have discussed the classification of the VisTex images, using intensity-based texture features (i.e., the co-occurrence matrix), color histograms, and a sequential of color and texture: the color correlogram. However, better classification results may be achieved when these methods are combined.

In the current section, a new color induced texture analysis approach is introduced: the parallel-sequential approach, which combines the color correlogram and the color histogram, as is visualized in Figure 10.3. This new approach is compared with the parallel texture analysis approach: the co-occurrence matrix combined with the color histogram, as is visualized in Figure 10.2.

First, the color histogram data and texture features were concatenated. The six best color histograms were used in combination with both the two best quantization schemes of each color space (for the color correlogram) and the best intensity quantization scheme (for the co-occurrence matrix). The RGB color histogram was excluded since it only performs well with a quantization that is computationally too expensive (see Table 10.2).

In Table 10.3, the results of the parallel approach (i.e., combination of color histogram and co-occurrence matrix, see also Figure 10.2) are provided. In general, the color histogram

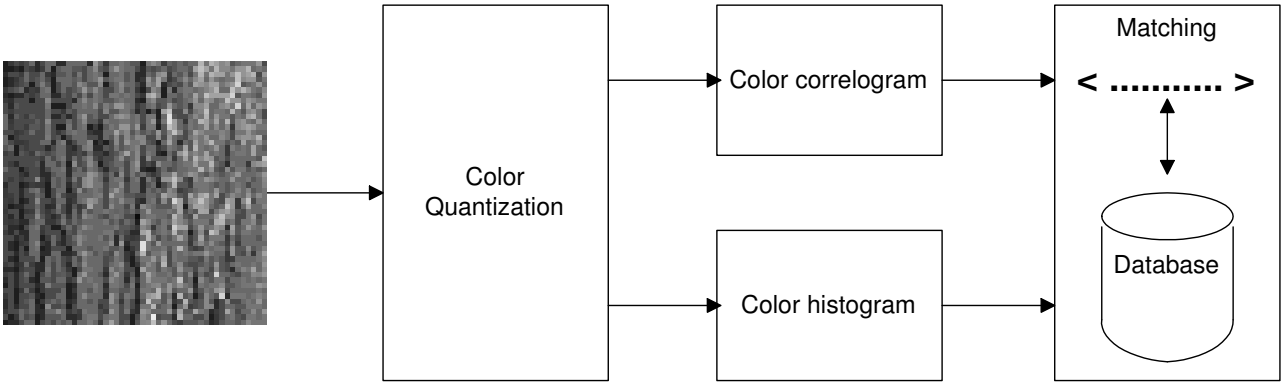


Figure 10.3: The new parallel-sequential approach for texture analysis which yields in parallel: global color analysis, using the color histogram, and color induced texture analysis, using the color correlogram.

based on the HSV 162 bins quantization scheme performed best ($91 - 92\%$). However, the computationally much cheaper 11 color quantization scheme did also have a high performance (88%), when combined with the on HSV 32 bins based co-occurrence matrix (see Table 10.3). Therefore, the latter combination should be taken into account for real-time systems, using color and texture analysis.

The new parallel-sequential approach has a correct classification ranging from 84% to 96% (see Table 10.3). So, the combination color histogram with color correlogram improved the classification performance significantly, compared to each of them separately (cf. Table 10.2 and 10.3).

The configurations using coarse color quantizations for the definition of the color correlogram, outperformed the more precise color quantizations for all color spaces. The 11 color categories color quantization using 27 bins for the color correlogram, performed best on average (92.6%), followed by the HSV-27 bins configuration (91.0%). Concerning the color histogram configurations, the highest average correct classification was provided by the HSV-162 bins color histogram (92.8%), followed by the 11 color categories color histogram with 91.2%.

The best color correlogram - color histogram combinations were: the 11 colors, 27 bins correlogram & 11 colors histogram, the 11 colors, 27 bins correlogram & HSV-162 color histogram, and the XYZ, 27 bins correlogram & LUV-125 color histogram (the percentages are denoted bold in Table 10.3). When considering the computational complexity of these combinations, the first combination should be preferred, with its feature-vector of size 15: 11 colors + 4 features derived from the 11 colors 27 bins color correlogram, as described in Section 13.4.

Table 10.3: The classification results of *the best combinations* of color histograms with co-occurrence matrices (the parallel approach, see also Figure 10.2) and with color correlograms (the parallel-sequential approach, see also Figure 10.3), using several quantizations of color spaces.

		Color histogram				
		11 colors	HSV-27	HSV-162	LUV-64	LUV-125
Co-occurrence matrix	HSV-32	88%	90%	92%	82%	90%
	LUV-8	84%	89%	92%	82%	88%
	RGB-8	84%	89%	92%	82%	88%
	XYZ-64	87%	84%	91%	79%	90%
	YUV/YIQ-8	83%	87%	92%	81%	89%
Color correlogram	11 colors-27	**94%**	92%	**96%**	92%	89%
	HSV-27	93%	87%	92%	92%	91%
	LUV-27	90%	89%	91%	88%	89%
	RGB-8	92%	91%	93%	86%	87%
	XYZ-27	87%	89%	92%	84%	**94%**

10.5 Conclusion

Determining the optimal configuration for color-based texture analysis is very important since the success of image classification and image retrieval systems depends on this configuration. Therefore, in this chapter, a series of experiments was presented exploring a variety of aspects concerning color-based texture analysis. The color histogram, the co-occurrence matrix, the color correlogram, and their combinations (i.e., the parallel and sequential approach) were compared with one another, using several color spaces and quantization schemes. A new texture analysis method: the parallel-sequential approach, was introduced.

The worst classification results were obtained when only intensity-based texture analysis (i.e., the co-occurrence matrix) was used, the best classification performance in this setting was 58% for the HSV and CIE LUV color spaces. Including color sequentially, using the color correlogram, gave better results (74%). The parallel approach (i.e., color histogram combined with the co-occurrence matrix improved the performance substantially (see Table 10.3). However, by far the best classification results were obtained using the new parallel-sequential approach (i.e., color histogram and color correlogram combined, a performance of 96% correct classification was obtained, using the HSV 162 bins color histogram in combination with the color correlogram for the 11 color categories with 27 bins. These results indicate that the use of color for image analysis is very important, as classification performance was improved by 38%, compared with the most widely used, intensity-based, co-occurence matrix. Moreover, in general, coarse color quantization schemes perform excellent and should be preferred to more precise schemes.

The success of the parallel-sequential approach emphasizes the importance of both the global color distribution in images, as identified by the color histogram, and the importance of the utilization of color with the analysis of texture. As was shown, ignoring color in either texture analysis or as a global feature impairs the classification of image material substantially. Moreover, the complementary character of global color and color induced texture analysis is illustrated.

Follow-up research should challenge the parallel-sequential approach, by exploring and comparing different texture analysis methods with the parallel-sequential approach introduced in this chapter. Moreover, the use of combining texture analysis methods should be investigated since it might provide the means to increase classification results [257]. Preferably, this research should be conducted using a much larger database of textures.

Regardless of texture analysis methods, note that the computationally inexpensive and well performing 11 color categories are human-based. In the next chapter, we will investigate whether the texture analysis techniques discussed in the current chapter can mimic human texture classification. This is of the utmost importance as it is the human who will use and judge the systems in which texture analysis techniques are incorporated [40, 270].

11

Mimicking human texture classification

Abstract

An attempt was made to mimic human (colorful) texture classification by a clustering algorithm. As test set, 180 texture images (both their color and gray-scale equivalent) were drawn from the OuTex and VisTex databases. First, a k-means algorithm was applied with three feature vectors, based on color/gray values, four texture features, and their combination. Second, 18 participants clustered the images, using a newly developed card sorting program. The mutual agreement between the participants was 57% and 56% and between the algorithm and the participants it was 47% and 45%, for resp. color and gray-scale texture images. Third, in a benchmark, 30 participants judged the algorithms' clusters with gray-scale textures as more homogeneous then those with colored textures. However, a high interpersonal variability was present for both the color and the gray-scale clusters. So, despite the promising results, it is questionable whether average human texture classification can be mimicked (if it exists at all).

This chapter is almost identical to:

Rikxoort, E. M. van, Broek, E. L. van den, and Schouten, Th. E. (2005). Mimicking human texture classification. *Proceedings of SPIE (Human Vision and Electronic Imaging X), 5666*, 215-226.

11.1 Introduction

Most computer vision (CV) and content-based image retrieval (CBIR) systems [30, 125, 193, 209, 270] rely on the analysis of features such as color, texture, shape, and spatial characteristics. Some of these CV and CBIR systems are partly founded on principles known from human perception. However, these systems are seldomly validated with experiments where humans judge their artificial counterparts. This, despite the mismatch that is present between user needs and current image retrieval techniques [117]. The current chapter discusses the process of such a validation for the artificial analysis of texture; i.e., mimicking human texture classification. In addition, the influence of color on texture classification is a topic of research.

As feature for the human visual system, texture reveals scene depth and surface orientation; moreover, it describes properties like the smoothness, coarseness, and regularity of a region (cf. Rao and Lohse [218], Battiato, Gallo, and Nicotra [9], Koenderink et al. [148], Pont and Koenderink [215], and Van Rikxoort and Van den Broek [223]). Texture is efficiently encoded by the human visual system; as Bergen and Adelson [14] stated: "... simple filtering processes operating directly on the image intensities can sometimes have surprisingly good explanatory power." Inspired by human texture processing, artificial texture analysis techniques describe similar properties as human perception does. However, direct comparisons between human and artificial texture processing are seldomly made.

In 2000, Payne, Hepplewhite, and Stonham [207] presented research toward mimicking human texture classification. Given a target image, they asked 30 humans to classify textures. Next, they compared these classifications with the classifications done by several texture analysis techniques. They concluded that, where the human visual system works well for many different textures, most texture analysis techniques do not. For only $20\%-25\%$ of the textures, a match was found between artificial and human classification.

The research of Payne et al. [207] concerned gray-scale texture analysis, as most research in CV and CBIR. This, despite that most image material is in color. As Palm [205] already denoted: "The integration of color and texture is still exceptional." From a scientific point of view, one can argue that since neither texture nor color is fully understood, the influence on each other is simply too unpredictable to do research in, at least outside a controlled experimental environment.

Color on its own, already is a complex phenomenon, as is texture. The perception of color is influenced by both environmental issues (e.g., position of the light source and properties of material) and internal processes present in the observer (e.g., color constancy [321]). However, concerning color classification or categorization, evidence is present for the existence of 11 color categories [16, 23, 28, 75, 219, 286]: black, white, red, green, yellow, blue, brown, purple, pink, orange, and gray, used by human memory (see Chapters 3–10). Re-

cently, this concept was embraced and utilized for the development of color-based image retrieval, as discussed in Chapters 7 and 8.

Our approach to mimicking human (colorful) texture classification is different from the approach used by Payne et al. [207] First, we let a k-means algorithm cluster the whole dataset using different feature vectors (Section 11.3). Next, in Section 11.4, we let humans cluster the whole dataset and in Section 11.5, we determine to which extend our k-means clusterings mimic the human clustering. As a follow up study, we let humans judge the clusters generated by the artificial clustering techniques (Section 11.6). We conclude with a discussion in Section 11.7.

11.2 Experimental setup

In this section, general specifications are provided, which hold for all three experiments: automatic clustering, human clustering, and humans judging the automatic clustering. As data, a collection of 180 colorful texture images were drawn from the OuTex [200] and VisTex [178] databases. Two criteria were used when selecting the images: (i) there had to be images from at least fifteen different categories and (ii), when a class was extremely large compared to the other classes, only a subset of the class is used. Moreover, the images were resized in order to fit on one screen. This was needed to facilitate an optimal and pleasant execution of the experiment. Figure 11.1 provides an overview of all the 180 images.

In both the first and the second line of research, two experiments were conducted: one with the original color images and one with gray versions of the same images. To obtain the latter, the set of 180 images was converted to gray-scale (I) images; see Equation 9.3. Now, two identical sets of images were present, except for presence versus absence of color information.

Clustering of images can be seen as sorting the images in a number of categories or stacks. So, the clustering of texture images can be treated as a card sorting task [190]. In such a task, the participant is asked to sort cards (e.g., images) and put them on separate stacks. As a consequence, only the top image on each stack is visible. So, participants have to memorize a representation of each of the stacks they defined. However, during the task the number of images on the stacks will increase and the content of the stack will change. Therefore, also the representation of the stacks needs to be updated, for which the human visual Short Term Memory (vSTM) [48] has to be taken into account.

Human vSTM can contain four [319] to fourteen [220] items. The number of clusters made by humans needs to be within this range. To be able to compare the clusters of textures made by the participants, they all had to define the same number of clusters. Moreover, the automatic clustering also had to result in the same number of clusters in order to be able to

Figure 11.1: An overview of all 180 images (the color version) used in the clustering experiments with both human participants and the automatic classifier. See Figure B.8 in Appendix B for a large color print of this screendump.

compare it with its human counter parts.

To determine this number of clusters, we asked five experts to cluster the images in an arbitrary number of clusters, with an upper limit of fourteen. The mean number of clusters produced by the experts is taken as the number of clusters to be produced. The experts determined the optimal number of clusters for this dataset, on both the gray-value and colorful images, to be six.

11.3 Automatic texture clustering

Automatic texture clustering is done in three steps, for both sets of images: (1) defining a suitable feature space, (2) calculate the feature vector of each image, such that each image is represented by a point in the feature space, (3) find groups or clusters of points in the feature space.

11.3.1 Clustering techniques

Many approaches have been developed for clustering points in feature space; see Mitchel [128] and Berkhin [15] for recent surveys. These approaches can be divided in two groups from the perspective whether or not additional information on data points is available.

Supervised approaches need, at least for a representative sample of the data points, information to which cluster each data point belongs. In our case this would mean dividing the data set provided by the human clustering into two or three parts: a part used for training a supervised method, a part for evaluating the parameters used during training (this is often not done as the available supervised data set is usually small), and a part for evaluating the final result of the clustering. In our case, the data set is too small to allow splitting it into parts.

Unsupervised methods do not need labeling of the data points. But usually they require the number of clusters as additional input. Either they use it as a fixed a priori number needed to start the clustering process, or they use it as a termination condition, otherwise they would continue until each data point is its own cluster. In our case, the number of intrinsic clusters, was determined by experts (Section 11.2), who determined the output to be six clusters. This enables us to compare the automatic clustering to the human clustering.

Since we did not have any information on the distribution of the points in our feature space, we evaluated two general applicable and often used methods: hierarchical clustering and k-means clustering. Evaluation of these two methods on an early available subset of our data did not show a preference for one of the two: the results were comparable. For this chapter, we chose to use the k-means method as it has somewhat more possibilities to tune certain parameters.

11.3.2 Feature vectors

In this research, three distinct feature vectors are used for the k-means algorithm. In Chapter 10, we determined the optimal configurations for both colorful and gray-scale texture classification. The optimal configuration for colorful texture analysis turned out to be our new parallel-sequential approach, using four texture features (i.e., entropy, inverse difference moment, cluster prominence, and Haralick's correlation), from the color correlogram [118] based on the 11 color categories [16, 23, 28, 75, 219, 286] combined with the 11 color histogram. For gray-scale texture analysis, the parallel approach performed best, in which the four texture features from the co-occurrence matrix [106, 257, 300] based on the HSV color space using 32 bins, are combined with a histogram from the HSV color space quantized in 27 bins.

In this experiment, for both color and gray-scale, k-means clustering was applied using three different feature vector configurations consisting of: (i) color or gray-scale information; i.e., the histogram, (ii) textural information; i.e., the four texture features, and (iii) both color and texture information; i.e., the histogram and the four texture features.

For each of the six vectors used in the k-means clustering, six clusters of images resulted. In Table 11.1, the size of each of the clusters is shown.

Table 11.1: The size of the six clusters constructed by the k-means algorithm for the different feature vectors for both color and gray-scale.

Feature vector	Color						Gray-scale					
Texture features	17	18	68	13	15	49	3	19	66	20	43	29
Color/gray-scale features	29	29	30	25	29	38	25	33	13	18	38	53
Combined features	42	25	24	25	28	36	15	14	49	28	32	42

11.4 Human texture clustering

11.4.1 Method

Eighteen subjects with normal or corrected-to-normal vision and no color deficiencies participated. They all participated on a voluntary basis. Their age ranged from 16 to 60. Half of them were male and half of them were female. All participants were naive with respect to the goal of the research and one of them was specialized in color or texture perception.

The experiments were executed on multiple PCs. In all cases the screen of the PC was set on a resolution of 1024×768 pixels. Moreover, we assured that the experiment was conducted in an average office lighting. We chose for this loosely controlled setup of apparatus, since it represented an average office situation and our opinion is that good algorithms mimicking human perception should be generally applicable and robust enough to handle images, which are taken and viewed under various circumstances. To put it in a nutshell, we consider the world as our experimental environment.

Two experiments were conducted. They differed only with respect to the stimuli; i.e., the texture images (see Section 11.2 and Figure 11.1). In one of the experiments color images were presented; in the other experiment their gray-scale equivalents were presented (see also Section 11.2). In order to control for possible order effects, half of the participants executed the experiments in the one order and the other half in the other order.

As discussed in Section 11.2, clustering of images can be represented as a card sorting task. However, in order to control and automate the sorting task as much as possible, a Tcl/Tk program was used that fully operationalized the desktop metaphor. A canvas

(i.e., window) was presented on a computer screen in which the images can be moved and stacked on each other, just like on a regular desktop [46, 96].

At the start of the experiments, the images are shown as a pile on the canvas. To tackle possible effects in sorting due to the order of presentation of the images, the images were placed in random order on the pile. So, at the start of the experiment, the canvas presented one pile of 180 randomly sorted images, as is shown in Figure 11.2.

The participants were able to drag the images by way of a mouse. They were allowed to drag the images all over the screen and drop them on any position wanted. During the experiment, all images were free to be positioned otherwise and, so, it was possible to change, merge, or divide already defined stacks. The only restriction was that the six resulting stacks were placed clearly separately from each other in order to tackle possible overlap between stacks. An example of such a final cluster is provided in Figure 11.2.

The participants were not instructed what features to use for the classification. This loose instruction guaranteed an unbiased human texture classification. The latter was of the utmost importance since we wanted to mimic human texture classification and were not primarily interested in the underlying (un)conscious decision making process.

After a definite choice of clusters was determined, the result was saved (by pressing the save button). For each image, its coordinates as well as its name were saved. Hence, the stacks could be reproduced, visualized, and analyzed easily.

11.4.2 Data analysis

For both experiments, the same data analysis was applied. In this section, the data analysis is described; in the next section, the results are presented.

For each of the 153 $(18!/(16!\cdot 2!))$ unique pairs of participants (p_i, p_j), a consensus matrix $(M_{(p_i,p_j)})$ of size 6×6 was determined, which contains for each pair of clusters, the number of images that match. Non-unique pairs of clusters were chosen since one cluster of participant i can encapsulate the images assigned to two separate clusters by a participant j and vice versa. From the set of confusion matrices, two data were derived: (i) the average consensus on the clustering between participants and (ii) the most prototypical set of clusters; in other words, the most prototypical participant.

The average consensus in the clustering between participants was determined as follows: For each pair of participants (p_i, p_j), the consensus $C_{(p_i,p_j)}$ is determined by summing the highest value of each of the six rows of the consensus matrix $M_{(p_i,p_j)}$, where each of the six values of each of the rows denotes the intersection (or consensus) between two clusters of p_i and p_j. So, $C_{(p_i,p_j)} = \sum_{i=1}^{6} max\{row_i\}$. Now, the overall consensus can be determined by: $\frac{1}{153}\sum_{p_ip_j} C_{(p_ip_j)}$.

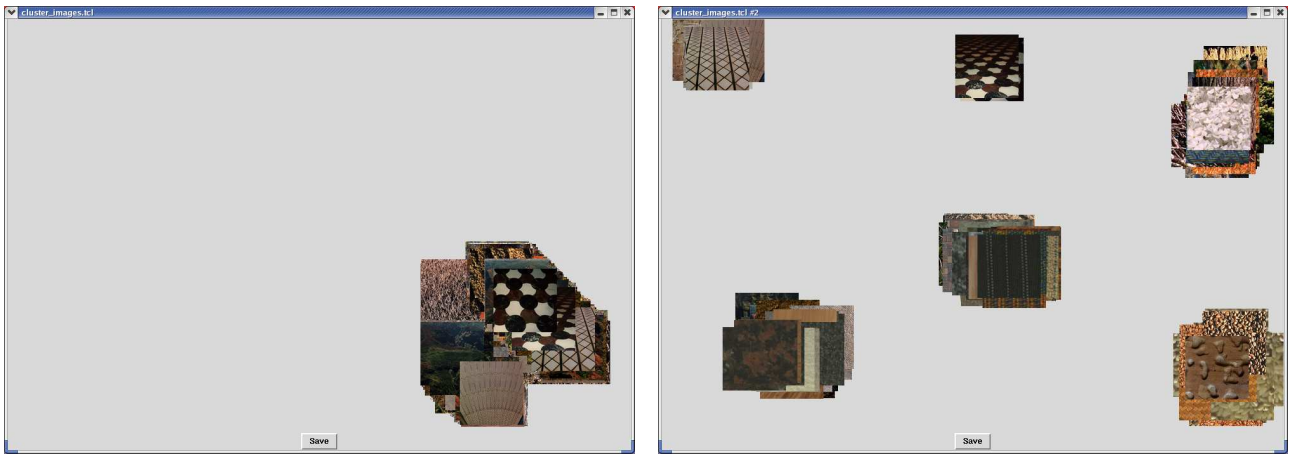


Figure 11.2: Above: The start condition of the experiment: one pile of 180 images. Below: An example of a final result of an experiment: six clusters of images. See Figure B.9 in Appendix B for a large full color print of these screendumps.

Of particular interest is the most prototypical set of clusters since it describes the most prototypical human (clustering); i.e., the highest average consensus to all other participants. The average consensus A of participant p_i is defined by: $A_{p_i} = \frac{1}{17} \sum_{j=1}^{17} C_{(p_i,p_j)}$. Subsequently, the most prototypical participant (or the most prototypical set of clusters) is defined as: $C_{ave} = max\{A_{p_i}\}$.

11.4.3 Results of colorful texture clustering

The average consensus between the participants with respect to colorful textures was 57%, ranging from 39% to 87%. The consensus matrix describing the consensus between all pairs of participants (Table 11.2, the numbers in a normal font), illustrates the variance present between the participants, in the clusterings. Please note, for color-based similarity judgments of images, participants also vary strongly, as described in Chapter 8.

In order to establish the most prototypical set of clusters, we determined a set of core images, based on the consensus matrices. For each consensus matrix (of two persons), for each cluster of p_i is determined which cluster of p_j matches best. Now, the core images are determined as follows: For each image is determined how often it is in the intersection of two clusters (of two distinct persons). Note, this interval ranges between 0 and 153. An image is labeled to be a core image when at least 45% (of 153 pairs) of the participants agreed that it is in a certain cluster. This approach is adopted from Payne et al. [207]. The threshold of 45% was chosen because the clustering is probably a fuzzy one and with this treshold, images can be assigned to two different clusters and still be a core image. For the colorful textures, this resulted in a set of 88 (out of 180) core images. The overall, average consensus between the participants on the core images was 70%.

Table 11.2: The consensus on the clustering between the 18 participants (p). The numbers in a normal font denote the colorful images; the numbers in the italic font denote the gray-scale images.

	p01	p02	p03	p04	p05	p06	p07	p08	p09	p10	p11	p12	p13	p14	p15	p16	p17	p18
p01	–	66	60	68	54	39	50	63	46	79	50	53	64	50	51	54	69	49
p02	*50*	–	67	74	50	39	50	56	51	78	48	45	58	44	49	56	46	60
p03	*63*	*55*	–	60	60	40	48	60	45	81	50	50	61	47	51	56	58	54
p04	*54*	*53*	*56*	–	50	38	49	70	54	87	53	55	67	44	57	60	63	61
p05	*48*	*51*	*50*	*42*	–	36	40	55	40	68	45	44	49	46	55	52	51	54
p06	*46*	*37*	*41*	*40*	*40*	–	38	49	45	70	42	41	49	39	46	42	41	37
p07	*54*	*58*	*78*	*53*	*50*	*43*	–	63	55	87	52	50	59	54	50	55	52	53
p08	*58*	*59*	*71*	*59*	*53*	*46*	*68*	–	56	83	54	54	56	49	50	53	62	50
p09	*52*	*51*	*65*	*47*	*46*	*36*	*60*	*53*	–	78	49	53	59	42	52	51	49	59
p10	*46*	*46*	*56*	*43*	*49*	*42*	*53*	*45*	*47*	–	51	51	54	41	54	54	51	51
p11	*68*	*55*	*70*	*63*	*50*	*54*	*63*	*66*	*64*	*64*	–	55	63	52	54	61	59	61
p12	*52*	*55*	*63*	*61*	*51*	*44*	*57*	*57*	*48*	*46*	*49*	–	55	47	49	50	50	54
p13	*60*	*45*	*56*	*55*	*43*	*41*	*47*	*49*	*45*	*49*	*64*	*58*	–	51	54	59	72	64
p14	*55*	*54*	*66*	*51*	*48*	*44*	*65*	*61*	*45*	*51*	*58*	*56*	*59*	–	50	56	49	59
p15	*47*	*53*	*57*	*47*	*60*	*49*	*53*	*54*	*51*	*49*	*47*	*52*	*50*	*56*	–	47	45	45
p16	*47*	*50*	*55*	*51*	*45*	*40*	*49*	*49*	*41*	*51*	*49*	*56*	*62*	*51*	*51*	–	52	65
p17	*65*	*53*	*60*	*61*	*45*	*40*	*51*	*55*	*41*	*40*	*51*	*61*	*74*	*50*	*44*	*49*	–	48
p18	*58*	*58*	*70*	*57*	*53*	*45*	*59*	*60*	*53*	*56*	*55*	*64*	*64*	*61*	*55*	*60*	*63*	–

Based on the set of core images, it was possible to determine the most prototypical participant. The participant with the highest average consensus with all other participants on these core images is the most prototypical participant. One participant did have an average consensus of 82% with all other participants; hence, the clusters of this participant are labeled as prototypical clusters.

The prototypical clusters are now used to determine the base images for all prototypical clusters. An image is be a base image for a particular cluster if it is assigned to the cluster by at least 8 $(18/2 - 1)$ participants. The clusters can be described by respectively, 37, 26, 14, 37, 37, and 45 base images. Moreover, 24 images appeared to be a base image for more than one cluster. The mean frequency of the base images in the clusters is 11.74.

11.4.4 Results of gray-value texture clustering

The average consensus between the participants with respect to gray-value textures was 56%, ranging from 36% to 78%. In Table 11.2, the numbers in an italic font, provide the consensus between all pairs of participants. As Table 11.2 illustrates, a considerable amount of variance is present in the consensus between the participants on the clustering of gray-scale textures.

For the determination of the core images, again a threshold of 45% was chosen. This resulted in a set of 95 (out of 180) core images, which is slightly more than with the color textures. In contrast, the average consensus between the participants on the gray-value images was slightly less (65%) than with the color textures. The participant assigned as prototypical, did have an average consensus of 73% to all other participants.

The clusters can be described by respectively, 32, 21, 32, 44, 24, and 46 base images. Moreover, 42 images appeared to be a base image for more than one cluster. The mean frequency of the base images in the clusters is 12.01.

11.5 Automatic versus human texture clustering

Since the goal of this research is to mimic human texture classification, we want to compare the automatically generated clusters to the clusters generated by the human participants. For this purpose, the same analysis is applied for the colorful textures and the gray-scale textures. For both the clusters of color and gray-scale images, each of the $54(18 \cdot 3)$ unique pairs of participant - automatic clusterings, a consensus matrix was constructed, using the base images (see Section 11.4.3). The base images were used since they describe the human clusters. Two types of similarity were derived from these matrices: (i) the overall consensus between the automatic clusterings and the human clusterings and (ii) the consensus based on the clusters defined by their base images (see also Section 11.4.3).

11.5.1 Data analysis

The consensus between the automatic and the human clusterings was determined as described in Section 11.4.2 with (p_i, p_j) being a pair of participant - automatic clustering, instead of a pair of participants. Next to the average consensus, the consensus on the prototypical clustering (as described in Section 11.4.3) is of interest. For this purpose, we will now define: a binary measure and a weighted measure.

11.5.1.A Binary measure of agreement

The binary measure of agreement assigns one cluster (c) to each image (I) by means of the frequency of assignment by the participants (see Section 11.4.3). The cluster with the highest frequency of assignment is assigned to the image (I_c). This clustering is compared to the automatic clusterings for each image (I_a) in a binary way.

Let ϕ be the binary value assigned to each image. Then, for each image I, ϕ is 1 when $I_c = I_a$ and ϕ is 0 when $I_c \neq I_a$. The total binary agreement is now defined by $\sum_\phi$. Last,

the binary agreement for each cluster x is defined by $\sum_{\phi} |c = x$. The total binary agreement is normalized by dividing it by the number of images.

11.5.1.B Weighted measure of agreement

The weighted measure of agreement weights the agreement on the clustering and is based on the frequencies of assignment to a cluster by humans. The frequencies are divided in four categories, the first category has a frequency of at least 15, the second category has a frequency of at least 11, the third category has a frequency of at least 7, and finally the fourth category has a frequency less than 7.

Let θ be the weighted measurement value for each image. Then, for each image I, θ is 3 when I_a is in the first category, θ is 2 when I_a is in the second category, θ is 1 when I_a is in the third category, and θ is 0 when I_a is in the last category. The total weighted agreement is now defined by $\sum_{\theta}$. The weighted agreement for each cluster x is defined by $\sum_{\theta} |c = x$. The weighted agreement is normalized by dividing it by the total weighted agreement of the most optimal clustering.

The weighted measure is used next to the binary (standard) measure because the human clustering is a fuzzy one and is only defined by the frequencies of assignment. In the binary measure, these frequencies are not used; hence, it can be considered as a baseline measure.

11.5.2 Results

11.5.2.A Colorful textures

For the colorful textures, three configurations (i.e., feature vectors) for k-means clustering were used (see Section 11.3): (i) the 11 color histogram, (ii) the four texture features, and (iii) a combination of the color and texture features, resulting in a feature vector of length 15.

For each of the three feature vectors, its average consensus with the participants' clusters was determined, as described in Section 11.4. The average consensus between human and automatic clustering using only color information was 45%, using only texture information it was 46%, and using both color and texture information it was 47%.

In Table 11.3, the results from the binary and weighted measures of agreement, between human and automatic clustering are given. It is possible that no images are assigned to a particular human cluster because we adopted the same approach for the calculation of the consensus as described in Section 11.4: non-unique mapping of the clusters. So, when one human cluster is matched twice by the artificial classifier, another cluster is not matched.

The percentages marked with a * in Table 11.3 are the result of the fact that no images were assigned to the particular cluster by the specific automatic clustering.

For the binary measure, there are two clusters on which one of the feature vectors had a percentage of more than 50%. For the weighted measure, four clusters present a consensus of more than 50% between human and artificial clusterings (see also Table 11.3).

Table 11.3: The percentages of agreement between human and automatic clustering in classification of the colorful images, for each cluster and for the whole dataset, using the binary measure and the weighted measure.

Cluster	binary measure			weighted measure		
	color	texture	combined	gray	texture	combined
1	35%	22%	32%	50%	50%	50%
2	42%	46%	42%	39%	62%	61%
3	0%*	0%*	0%*	100%*	100%*	100%*
4	22%	16%	22%	58%	73%	69%
5	76%	54%	60%	60%	83%	45%
6	40%	53%	53%	85%	43%	71%
All images	44%	39%	43%	42%	44%	45%

11.5.2.B Gray-scale textures

For the gray-scale textures, three configurations (i.e., feature vectors) for k-means clustering were used (see Section 11.3): (i) the 32 bins HSV gray-scale histogram, (ii) the four texture features, and (iii) a combination of the histogram and texture features, resulting in a feature vector of length 36.

For each configuration of automatic clustering, its average consensus with the participants' clusters was determined, as described in Section 11.4. The average consensus on the automatic clustering, using only gray-scale information was 44%, using only texture information it was 45%, and using gray-scale and texture information it was 42%.

In Table 11.4, the results from the binary and weighted measures of agreement, between human and automatic clustering are given. For the binary measure, there are four clusters on which one of the automatic classifiers had a percentage of more than 50%. For the weighted measure, five clusters present a consensus of more than 50% between human and artificial clustering.

Table 11.4: The percentages of agreement between human and automatic clustering in classification of the gray-scale images, for each cluster and for the whole dataset, using the binary measure and the weighted measure.

Cluster	binary measure			weighted measure		
	gray	texture	combined	gray	texture	combined
1	100%	44%	50%	97%	47%	62%
2	52%	0%	62%	100%	70%	59%
3	0%	0%	0%	100*%	100*%	100*%
4	61%	68%	68%	79%	65%	71%
5	88%	0%	83%	100%	100*%	100%
6	0%	7%	0%	100*%	100%	100*%
All images	36%	41%	44%	33%	41%	43%

11.6 Humans judging automatic clustering

As a follow up experiment, humans were asked to judge the clusters generated by the automatic clustering algorithm. For both color and texture, the clusters of the automatic clustering algorithm with the best average performance were chosen. For color, the k-means algorithm, using color and texture features was selected. For gray-scale, the k-means algorithm, using only texture features was selected.

For this experiment, the benchmark was used as was introduced in Chapter 6. It allowed users to judge each individual cluster for its homogeneity and correctness. The benchmark showed all images of a cluster in one screen, at the bottom of the screen a mark between 1 and 10 can be given for the homogeneity of the cluster shown. All users are presented with 12 screens: 6 containing gray-scale images and 6 containing colorful images. A screendump from the benchmark is shown in Figure 11.3.

In this experiment, 36 subjects, with normal or corrected-to-normal vision and no color deficiencies participated. Their participation was on a voluntary basis and they were naive with respect to the goal of the research. The age of the participants varied from 18 to 60, half of the participants were male and half of them were female. The experiment ran online. The participants were instructed to judge the clusters on their homogeneity. They were not informed about the clusters being produced by artificial classifiers.

For both the colorful and the gray-scale texture clusters, we determined the average rating given for the homogeneity of the results. The average rating for the gray-scale clusters was 6.1, with a standard deviation of 3.1; the average rating for the colorful clusters was 5.2, also with a standard deviation of 3.1. The gray-scale clusters were judged significantly better than the colorful clusters ($p < .0069$). The high standard deviations of the ratings denote a high variation between the participants in judging the clusters.

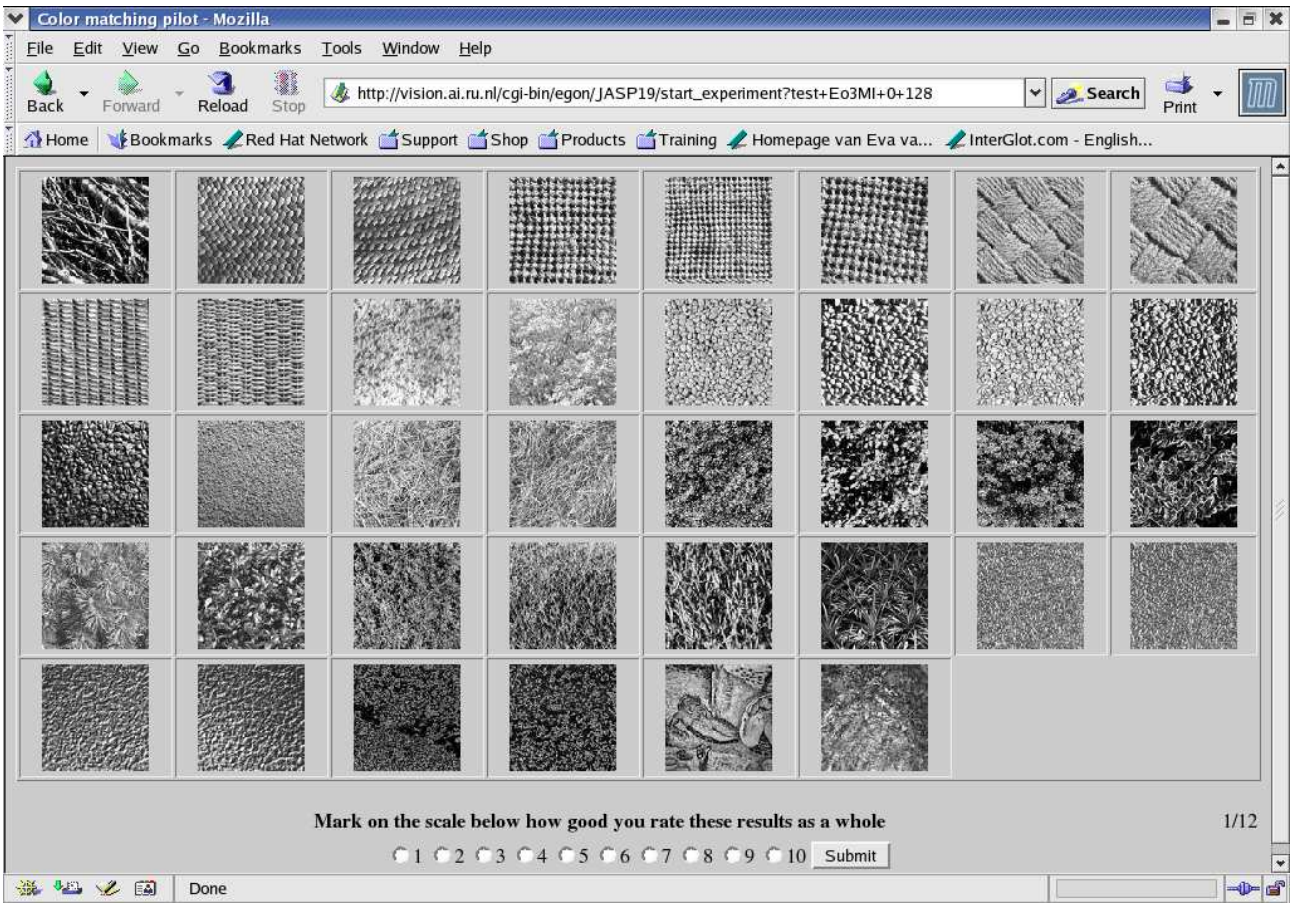


Figure 11.3: An example screen from the benchmark used to let users judge the automatic clusters. See Figure B.10 in Appendix B for a color print of this screendump.

11.7 Discussion

In the present research, first a set of 180 texture images were clustered by a k-means clustering algorithm, using three different feature vectors for both color and gray-scale. Next, 18 humans were asked to cluster the set of texture images both in gray-scale and color. Using the clusterings of all participants, a set of base images for each cluster was derived, which describe the clusters. The automatic clusterings were compared to the human clustering using two measures of agreement (i.e., binary and weighted). In addition, the influence of color compared to gray-scale was investigated. Last, a benchmark was executed in which 36 participants judged the automatic clustering results.

Note that in artificial versus human texture classification, a gap is present between the low-level textural features and human texture perception. By mimicking human texture classification, we aim at mimicking the outcome of human texture classification, we do not claim that our method mimics the process of human texture classification. To be able to truly mimic this process, fundamental research should be conducted, as was done for the development of the human-based, 11 color categories (see Chapter 3).

There is little literature available on human texture classification and the literature that is available uses gray-scale images and reports poor results [207]. One of the exceptions is the research of Rao and Lohse [218] who developed a texture naming system by asking humans to judge the properties of a set of textures. They concluded that there are three main characteristics of texture that are important in human texture vision: repetitivity, contrast, and coarseness. However, only 56 gray-scale textures were used in their research. To bridge the gap between the low-level textural features and human texture perception, such research should be extended to color and be conducted with a larger database, as was done in the current chapter.

For both the colorful and gray-scale textures, little consensus was present between the participants. Although all participants reported more trouble clustering the gray-scale images, the consensus between the participants was almost the same on the colorful textures and the gray-scale textures (57% vs 56%). The low consensus between the participants indicates that the task of clustering the textures selected was not a trivial one, as was our aim in selecting the images (see Section 11.4).

The overall success in comparing the automatic classifier to the human classifications was the same for the colorful textures and the gray-scale textures ($45\% - 47\%$ versus $42\% - 45\%$). However, when inspecting the results for the separate clusters, more success is shown on the gray-scale clusters. For the gray-scale textures, using the binary measure of agreement, for four clusters more than 50% of the images were classified correct. The weighted measure for the gray-scale images gives a good result on five of the clusters. The mean percentages of correct classification for the clusters, which are matched well, are 76% and 95% for the binary and weighted measure respectively.

For the colorful textures, there are respectively two and four clusters that match well. For the clusters which are matched well, the mean percentages of correct classification are 65% and 80%, for the binary and weighted measure respectively. So, the match in clustering between humans and the k-means algorithm is more convincing for the gray-scale images than for the colorful images. This effect of overall performance versus cluster-performance is caused by the non-unique mappings we used to determine the consensus between clusters. For gray-scale, there are six instances in which no images are assigned to a particular cluster (see Table 11.4) which impairs the results over the overall dataset. Moreover, for the clusters to which images are assigned, good to excellent results are obtained. For the colorful images, there are only three instances in which no images are assigned to a particular cluster, where the results for the other clusters are not convincing either.

An inspection of the images itself revealed that the clusters that are mimicked well by the automatic classifiers, show little variation in color/gray-scale and texture. So, all images in a well mimicked cluster, have the same texture properties like randomness, directionality, and coarseness, and show little variation in color/gray-scale. For both gray-scale and color,

the cluster to which no images were matched by the automatic classifiers, seem to be a 'garbage group', in which the human participants put all images they were unable to label. Such a 'garbage group' was mentioned by all participants. This group should be excluded from further analysis since the participants judged that the images in this group show no overlap in texture, color, or semantics.

That the gray-scale images are better mimicked by the automatic clustering methods can partly be explained by the trouble humans reported in clustering the gray-scale images. These difficulties in clustering were mainly caused by the fact that on the gray-scale images less semantic information is visible, due to the absence of color. So, on the gray-scale images humans use more pattern and gray-scale based clustering than semantic based clustering. In contrast, on the colorful images, most humans used semantic features for clustering.

Although human gray-scale texture clustering was better mimicked by automatic clustering, the results on colorful texture clustering were also satisfying. Especially when compared with other recent research such as that of Payne et al. [207] who reported a mean correct classification on gray-scale textures that was only $20\% - 25\%$. So, despite the low percentages of consensus between humans and the clustering algorithm, the results should be considered as promising. With that, this research presents a successful first attempt to mimic human colorful texture classification.

In future research, however, a four step approach should be adopted: (i) Let a group of participants cluster the images. (ii) Based on the clusters of this first group, a set of core images on which all participants agree to some extend can be determined. (iii) Last, a second group of participants should cluster this group of core-images. (iv) The resulting clusters should be compared with results from automatic clustering algorithms. Such an approach can, on the one hand, help in determining generic characteristics of human texture analysis and, on the other hand, a functional model can be generated, which mimics human texture classification.

Research toward human texture analysis and classification has just started. The current chapter did discuss one of the few attempts done so far in mimicking human texture classification. However, we have established an optimal configuration for automatic texture classification, as determined in Chapter 10 and showed that it is also successful in mimicking human texture classification. The next chapters will illustrate its use for applications like CBIR. Since the results of CBIR engines are judged by humans, better results can be expected from human-based techniques. In the next chapter, we will present the development of an Object-based Image Retrieval (OBIR) engine, which makes use of the optimal configuration for texture analysis to perform image segmentation and image retrieval.

12

The development of a human-centered object-based image retrieval engine

Abstract

The development of a new object-based image retrieval
(OBIR) engine is discussed. Its goal was to yield intuitive results for users by using human-based techniques. The engine utilizes a unique and efficient set of 15 features: 11 color categories and 4 texture features, derived from the color correlogram. These features were calculated for the center object of the images, which was determined by agglomerative merging. Subsequently, OBIR was applied, using the color and texture features of the center objects on the images. The final OBIR engine, as well as all intermediate versions, were evaluated in a CBIR benchmark, consisting of the engine, the Corel image database, and an interface module. The texture features proved to be useful in combination with the 11 color categories. In general, the engine proved to be fast and yields intuitive results for users.

This chapter is almost identical to:
Rikxoort, E. M. van, Broek, E. L. van den, and Schouten, Th. E. (2005). The development of a human-centered object-based image retrieval engine. In B. J. A. Kröse, H. J. Bos, E. A. Hendriks, and J. W. J. Heijnsdijk (Eds.), *Proceedings of the Eleventh Annual Conference of the Advanced School for Computing and Imaging*, p. 401–408. June 8-10, The Netherlands - Heijen.

12.1 Introduction

Humans differ in all imaginable aspects. This is no different for the characteristics of human vision. However, "the variance of human vision characteristics is much smaller than the gap between the characteristics of human vision and computer vision [261]". The latter is frequently called the semantic gap in computer vision and content-based image retrieval (CBIR) [31].

In order to bridge this semantic gap, the usage of appropriate prior knowledge is very important [234]. Ontologies, user preference profiles, and relevance feedback techniques were developed to utilize such knowledge. However, such methods require an enormous effort and consequently can only be applied in a limited domain [327]. We address the semantic gap from another angle, since we aim at developing techniques that are human-based and may lead to generic methods that was applied in an unlimited domain.

Our approach to improve the performance of CBIR systems is twofold: (i) we utilize knowledge concerning human cognition and (ii) we exploit the strength of image processing techniques. From this perspective, we aim to develop new image processing, classification, and retrieval techniques, which have low computational costs and provide intuitive results for users [187].

These techniques were inspired by human visual short-term memory (vSTM). Human vSTM can encode multiple features only when these features are integrated into a single object, defined by the same coherent boundary. Moreover, it has a storage limit between four items [319] and (at least) fourteen items [220]. Intrigued by the efficiency of human vSTM, we adapted a similar approach for our image analysis techniques.

In sharp contrast with human vSTM, in CBIR the features color and texture are most often analyzed over the complete images. However, with such an average description of images, a loss of information is present; i.e., characteristics of parts of images (e.g., objects) are lost. Moreover, most CBIR image processing schemes use large feature vectors; e.g., PBIR-MM (144 features: 108 color and 36 texture related) [155] and ImageRover (768 features) [255]. Since we aim to yield intuitive results for users [187] using computationally cheap methods, we mimicked the characteristics of the vSTM. Subsequently, we do not utilize complex shapes but applied a coarse segmentation algorithm, based on agglomerative merging [201], as described in Section 12.2. The content of the selected segments of images are compared with each other, using the highly efficient 11 color quantization scheme (see Chapters 3 and 5) and the color correlogram (see Chapters 9–11). This setup was tested in the newly developed CBIR benchmark (see Chapters 6–8) and adapted (see Section 12.4), resulting in a new CBIR engine. The performance of the final engine was measured (see Sections 12.5 and 12.6). Finally, in Section 12.7, a brief discussion can be found.

12.2 Image segmentation

The purpose of image segmentation is to divide an image into segments or regions that are useful for further processing the image. Many segmentation methods have been developed for gray level images and were later extended to color images; see Cheng, Jiang, Sung, and Wang [53] for an overview of them.

12.2.1 Segmentation by agglomerative merging

Segmentation was applied by agglomerative merging, as described by Ojala and Pietikäinen [201]. Their algorithm is a gray-scale image algorithm but was extended to color images using a color texture descriptor. The algorithm was applied using the color correlogram as texture descriptor that was based on the 11 color quantization scheme.

At the initial state of the agglomerative merging algorithm, the images were divided in sub blocks of size 16×16 pixels. At each stage of the merging phase, the pair of blocks with the lowest merger importance (MI) was merged. This merger importance is defined as follows:

$$MI = p \times L, \tag{12.1}$$

where p is the number of pixels in the smaller of the two regions an L is defined as:

$$L = |I - I'| = \sum_{i,j=0}^{m-1} |C^{\bar{d}}_{i,j}(I) - C^{\bar{d}}_{i,j}(I')|, \tag{12.2}$$

where m is the number of bins used and $C^{d}_{i,j}(I)$ is the color correlogram of image I (see Equation 9.1), and $\bar{d}$ is set to 1 (see Section 9.7.5 and 9.7.4 of Chapter 9). The closer L is to zero, the more similar the texture regions are. The agglomerative merging phase continues

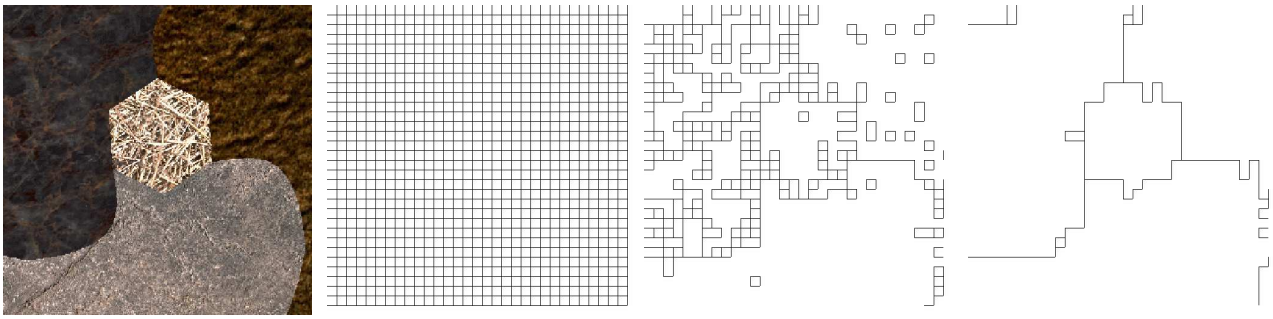


Figure 12.1: The segmentation process, from left to right: The original image, division of the image in blocks of size 16×16, the regions after 800 iterations of agglomerative merging, and the final segments. See Figure B.11 in Appendix B for larger full color prints of these images.

until the experimentally determined stopping criterion (Y), given in Equation 12.3 is met:

$$MI_{stop} = \frac{MI_{cur}}{MI_{max}} < Y, \tag{12.3}$$

where MI_{cur} is the merger importance for the current best merge, MI_{max} is the largest merger importance of all preceding merges. The agglomerative merging phase is illustrated in Figure 12.1.

12.2.2 Parameter determination

In order to use the segmentation algorithm, the parameter Y from Equation 12.3 had to be determined. This was done using a small test set of texture mosaics. In addition, three variations of the Merger Importance (MI), as given by Equation 12.1, were evaluated: (i) the form as given in Equation 12.1, (ii) $\sqrt{p}$ instead of p in calculating the MI value, and (iii) not using the number of pixels at all. The third variant showed to work best. In Figure 12.2 the behavior of the MI_{stop} value for the three merger importances (MI) are visualized. Using a sample set, the threshold Y (see Equation 12.3) was experimentally set on 0.6000.

With the introduction of the segmentation algorithm, all ingredients for an image description are defined: the color correlogram, the 11 color categories, and coarse image segmentation. Next, we will discuss the CBIR benchmark, which includes the CBIR engine, which uses the image description.

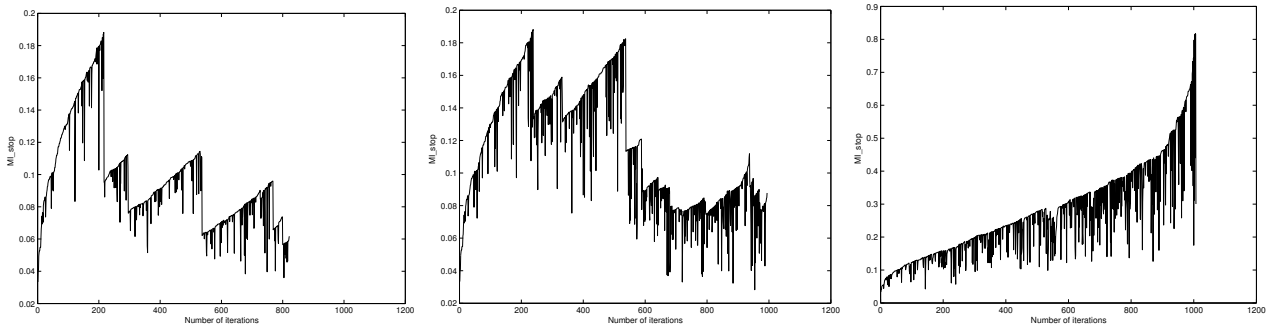
12.3 CBIR benchmark

In order to perform image retrieval using the image features discussed in the previous sections, a test environment or benchmark has been developed [31]. The three main components of this benchmark are: (i) The CBIR engine, (ii) an image database, and (iii) the dynamic interface module.

The CBIR engine calculates a feature vector for each image or image segment. Based on this feature vector, the distance between the query image and all other images is calculated by means of a distance measure. The result of this CBIR engine is a list of the top 100 most similar images to the query image. The most important parameters that can be set for the engine are: the distance measure and the feature vector.

Since the benchmark is modular, an image database of choice can be used. In principle, every database can be connected to the benchmark; the most common file-types are supported.

The dynamic interface module generates an interface in which the results can be presented. By way of a set of parameters, a range of options can be altered. For example, one



(a) (b) (c)

Figure 12.2: The MI_{stop} value (see Equation 12.3) for the merger importance: (a) $MI = p \times L$ (see Equation 12.1), (b) $MI = \sqrt{p} \times L$, and (c) $MI = L$, of the agglomerative merging phase, where p is the number of pixels in the smaller of the two regions and L is a distance measure.

can set the number of images presented for each query, the number of queries to be judged, and choose whether the presentation of the results is in random order or not.

For the present research, we have chosen as main settings: the intersection distance measure, the Corel image database, which is a reference database in the field of CBIR, and a presentation of the top 15 images retrieved in a 5×3 matrix, randomly ordered (see Figures B.12 and B.13 in Appendix B).

The histogram intersection distance (D) of Swain and Ballard [287] is used to calculate the difference between a query image(q) and a target image (t):

$$D_{q,t} = \sum_{m=0}^{M-1} \mid h_q(m) - h_t(m) \mid, \tag{12.4}$$

where M is the total number of bins, h_q is the normalized query histogram, and h_t is the normalized target histogram. This distance measure is developed for histograms but also works for texture feature vectors [322].

Three different feature vectors were used: (i) the histogram of the 11 color categories (see Chapters 3 and 5), (ii) the 4 texture features (see Chapter 9), and (iii) the color categories and texture features combined, resulting in a vector of length 15.

12.4 Phases of development of the CBIR engine

The final CBIR engine was developed in four phases. The final engine of each phases can be found online; see Table 12.1 for the web-address of each of the 13 benchmarks, including the final benchmark. The results of each benchmark (in each phase) were judged by two experts, who each judged 50 random chosen queries on the quality of the retrieved images.

Table 12.1: The addresses of the 13 different benchmarks, using either color, texture, or a combination of both features. The * stands for http://eidetic.ai.ru.nl/egon/. The final benchmark is indicated bold.

Phase	Color	Texture	Color and Texture
1	*/JASP1	*/JASP2	*/JASP12
2a	*/JASP19c	*/JASP19t	*/JASP19
2b	*/JASP29c	*/JASP29t	*/JASP29
3	*/JASP8catsC		*/JASP8catsCT
4	*/JASP8catsC-center		***/JASP-final**

12.4.1 Phase 1

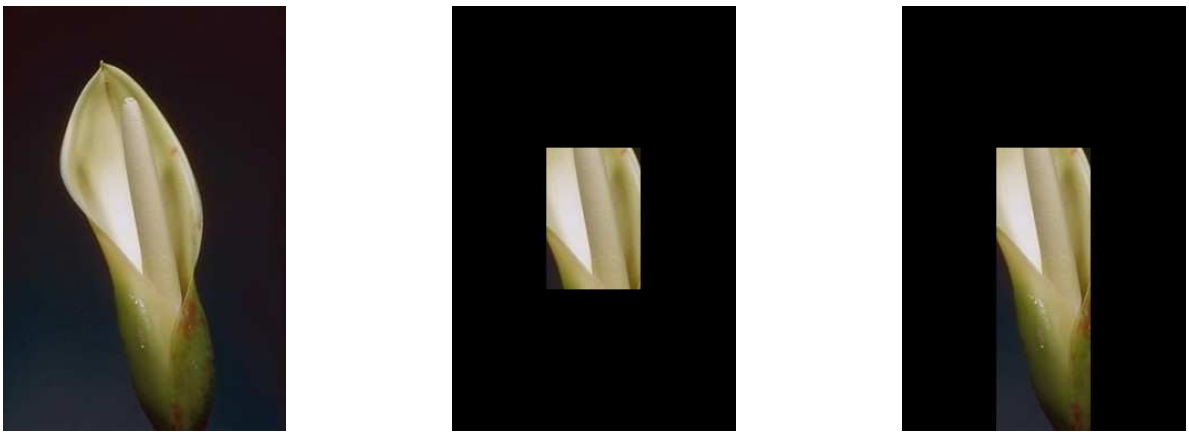
In the first phase of the development of the CBIR engine, the Corel image database (consisting of 60,000 images) was used as a test set. The segmentation algorithm, described in Section 12.2, was applied on each image in the database. Resulting segments were used for the CBIR engine if its area was more than or equal to 20% of the total area of the image; smaller ones were discarded.

People are, in most cases, interested in objects on the image [249]. Multiple objects can be present, not necessary semantically closely related (e.g., a person standing next to his car). So, one image can satisfy two unrelated queries (e.g., persons and cars). Hence, we have chosen to use each segment separately in searching the database of images.

In previous research on using texture based segmentation for CBIR, only one type of feature vector was chosen for the matching phase [322]. In a first attempt to apprehend the influence of texture in color image retrieval, three CBIR-engines were developed: a color-based, a texture-based, and a color&texture-based engine. With this approach we aim to evaluate the influence of texture features on the retrieval results.

Let us briefly summarize the results, as judged by the experts. The retrieval results of the color and of the color&texture-based engine were judged as being on an acceptable level. The results of the texture-based engine were very poor.

The inspection of the results revealed two problems: (i) The areas that exceeded the threshold of 20% did frequently form the background of the scene presented on the image and (ii) Frequently, no area exceeded the threshold of 20%. These two problems indicate that often we were not able to detect objects in the images. Therefore, in Phase 2, we will try an alternative method for segmentation.



(a) (b) (c)

Figure 12.3: (a) The original image. (b) The $\frac{1}{9}$ center grid cell of the image as used for analysis. (c) The $\frac{2}{9}$ center grid cells of the image as used for analysis. See Figure B.15 in Appendix B for large color versions of these three images.

12.4.2 Phase 2

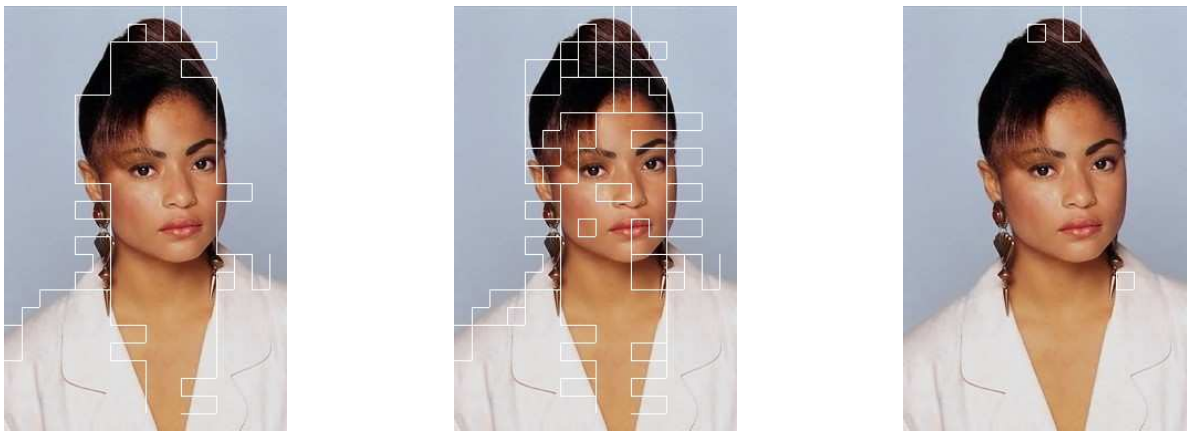
The making of most photos is initiated by the interest in certain objects. Therefore, the photographer will take care that an adequate presentation of the object(s) is present within the frame of the photo. In most cases, this means the object of interest is placed central in the photo. Thus, the central position of the image is of the utmost importance. This also holds for non-photo material: Imagine an image of a painting, of a sculpture, or of a cartoon. Also for this image material both the photographer as well as the artist who made the original, will place the object(s) in the center of the image.

Most images will present an object; but what to do with those images that present a scene (e.g., the sunrise on a photo or a landscape on a painting)? In such a case, the center of the image will not hold the object of interest but will hold a sample of the scene of interest. So, in one way or the other, the center of the image contains the most important information.

In order to investigate this hypothesis, we conducted a new research toward CBIR without image segmentation. We simply selected the center of the image. In order to do this, a grid of 3×3 grid cells was placed over the image. The center of the image was defined in two ways: (a) the center grid cell (see Figure 12.3b) and (b) both the center grid cell ans the cell below the center grid cell (see Figure 12.3c).

We were still interested in the influence of color, texture, and their combination (see Section 12.3). Hence, for each of the center definitions, three CBIR engines were developed, making a total of six CBIR engines developed in this phase (see also Table 12.1). The six engines retrieved their images from the complete Corel image database.

Similar to Phase 1, the engines relying on texture features solely performed poor. With



(a) (b) (c)

Figure 12.4: Segmentation of images with several parameters: (a) The correct parameter for its class (0.700). (b) The generic parameter as used in phase 1 (0.600). (c) The parameter of the class cats (0.800). See Figure B.14 in Appendix B for large color versions of these three images.

that, the evidence was strengthened that texture solely is not useful for CBIR. Hence, in the next phases of development, texture features on its own will no longer be used. For the color and color&texture-based engines, the center image approach proved to be successful. According to the experts, the $\frac{1}{9}$ approach performed slightly better than the $\frac{2}{9}$ approach. However, the results were still far from satisfying.

12.4.3 Phase 3

In this Phase, we aim to tackle the problems of segmentation, due to the variety of images in image classes. In order to tune the segmentation algorithm, the parameter Y (see Equation 12.3) had to be set separately for each class of images used. Except from tuning the parameters, the segmentation is similar to the segmentation in Phase 1. In this phase, similarity based on color and a combination of color and texture were used. Both engines were applied on seven classes of the Corel image database (i.e., cats, dogs, food, flowers, women, waterfall, and dinos), resulting in a database of 900 images. For each of these seven classes, the segmentation algorithm was applied using its own parameter.

As expected, tuning the segmentation algorithm for each class separately improved the retrieval performance substantially. The effect of tuning the segmentation algorithm for each class separate is illustrated in Figure 12.4. Furthermore, including texture features in the engine, improved the retrieval, compared to the retrieval results of the engine using color solely. However, the results were still not fully satisfactory; therefore, in phase 4, a combination of phase 2 and phase 3 is applied.

12.4.4 Phase 4: The final CBIR engine

Since both Phase 2 and Phase 3 provided promising results, we chose to combine both approaches: both the selection of the center of the image and the tuning of the segmentation for each class of images are utilized.

The procedure is as follows: (i) the image is segmented, (ii) the center grid cell is selected, and (iii) the region with the largest area within the segmented center grid cell is selected for analysis. So, for each image only one region represents the complete image. We assume that this region represents the object, which is the subject of the image. This process is illustrated in Figure 12.5.

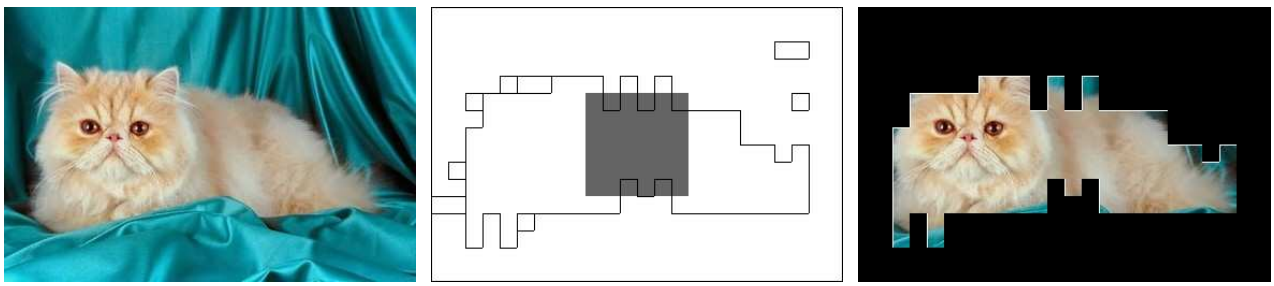
The results of both the color and the color&texture-based engine were promising. The color&texture-based engine performed better than the engine based on color solely. So, finally a successful setup was found and the final CBIR-engine was defined. In order to validate the success of the engines, we wanted to conduct a more thorough analysis of the retrieval results. This process of validation is described in the next two sections.

12.5 Measuring performance

12.5.1 Recall and precision

Two methods of validation can be applied in CBIR; both adapted from the field of Information Retrieval. Given a classified database with labeled images, recall and precision of the retrieval results can be determined. Recall signifies the percentage of relevant images in the database that are retrieved in response to the query. Precision is the proportion of the retrieved images that is relevant to the query.

In this experiment, it is not possible to determine recall of the system because the num-



(a) (b) (c)

Figure 12.5: (a) The original image. (b) The segments in the image and the grid placed on it. (c) The final region. See Figure B.16 in Appendix B for large color versions of these figures.

ber of relevant image are not known beforehand. A similar problem is present when querying the Internet. However, in both cases the precision of the system can still be determined.

In most CBIR research, precision is determined automatically, provided a well annotated database. However, with such an approach a problem arises with the Corel image database as it is used. The classification is done with only one keyword. As a result separate categories (i.e., categories labeled by different keywords) can have considerable overlap.

In order to tackle this problem with automatic determination of precision, we utilized a manual determination of precision. Recently, this approach was successfully applied [31, 187]. Users were asked to judge the retrieved images as either related to the query image or as not related.

To facilitate the manual determination of precision, the benchmark was utilized. The users were asked to judge the images retrieved by comparing them to the query image. The judgment was binary: either an image was judged as appropriate and selected, or an image was judged as inappropriate and not selected. For each query, the top 15 images, as determined by the CBIR engine, were presented to the users. To facilitate a rapid judgment of the query results, the query images were pre-defined; i.e., the user did not have to search for and select a query image, a random selection of query images was already taken from the database. For each query, we can then define the precision of the presented images. The number of 15 retrieved images is a compromise. It is low enough to allow all images of a query to be presented on one screen in a size that is suitable for judgment. This optimizes the speed of judgment and thus maximizes the number of queries that can be judged.

12.5.2 Semantic and feature precision

In everyday life, search-engines are judged on their semantic precision; i.e., do the results have the same meaning as the query? However, two possible problems arise: (i) the query is ill defined or (ii) the search engine's algorithm are not able to interpret the query correct. The interest in this distinction lays in whether the user or the engine can be blamed.

In CBIR the same problems arise. However, since the field of CBIR is young relative to that of (text-based) Information Retrieval and its techniques are not fully grown, the problems have a larger impact on the judged semantic precision. However, it is not yet possible to search on semantics; it is done through the features that correlate strongly with semantic categories.

Frequently, users do not understand the results a CBIR query provides, when they are naive to the techniques behind the CBIR engine. For example, the query image can contain a dog with brown hair. The CBIR engine can return other dogs with brown hair (e.g., see Figure B.12 in Appendix B), but also cats with a brown coat and women with much brown

hair. From a semantic point of view, the latter latter two results are incorrect; however, from a feature point of view, one can perfectly understand them.

We asked a group of eight users, who participated in judging the two CBIR engines, to judge the engines twice: once on semantic precision and once on precision based on features. In the next section, we will discuss the results of both of these judgments, for both engines, in general and for each class separately.

12.6 Results

In Section 12.4.4, the final color and color&texture-based engines were introduced. They use 11 color and 4 texture features of the center segment of each image. Since one of our main interests was whether or not texture features contribute to the correct classification and retrieval of images, both engines had to be judged by users.

In addition, in the previous section we have explained our interest in the difference between semantic precision and feature-based precision. For the latter judgments, the eight participating users were instructed to judge the retrieved images on the similarity with the query image, based on the patterns present (e.g., grass, hair, clouds) and on the color distributions.

These two differentiations result in four different situations in which precision of retrieval had to be determined. In total, the eight users judged 640 queries (20 per person per situation) and so provided a manually determined precision. The precision was determined over the top 15 matches of the queries, by selecting the images that are considered to be correctly retrieved (see also Section 12.5.1).

For each situation we determined the average number of selected images and with that the precision of the engine for each situation (see Table 12.2). Both the precision on feature level ($p < 0.0286$) and the precision on semantic level ($p < 0.0675$) is higher for the color&texture-based engine (feature: 8.51; semantic: 6.91) than for the color-based engine (feature: 7.39; semantic: 6.14).

In other words, no matter from which perspective the engines were judged, texture increased the precision of the retrieval performance. In addition, note that when the engines were judged on semantics significantly less images were selected then when judged on image features (color: $p < 0.0036$ and color&texture: $p < 0.0016$; see Table 12.2).

We will now present the average and standard deviation of the number of selected images for each of the seven classes separate, for each of the four situations (see Table 12.3). A large variance between the classes becomes apparent. The average number of images selected, for the seven classes, in the four situations, ranges from 2.20 (food; color-based,

Table 12.2: The average number of images selected when judging feature and semantic precision. The p values (determined by a two-tailed Student's t-test) indicate the difference between using only color features and using color and texture features as well as the difference between when judging feature-based or on semantic precision.

	color	color-texture	p value
feature	7.39	8.51	0.0286
semantic	6.14	6.91	0.0675
p value	0.0036	0.0016	

semantic) to 11.89 (dinos; color&texture-based, feature). Additionally, within most classes a considerable variability is present, as indicated by the standard deviations presented in Table 12.3.

Please note that all classes used are object-classes, except the class food. This class represents a concept on another level of semantics. The class food contained, for example, images of: plates with food on it, a champagne glass, people eating, and a picnic setting with a boat in a lake as background.

A class as heterogeneous as food, is impossible to classify with a high semantic precision. This is sustained by the poor results: 2.20 (color) and 2.85 (color& texture) images selected per query. In addition, the class food was the only class for which the use of texture substantially reduced the precision of retrieval. For the class flowers texture did decrease the precision of retrieval as well, but to a lower extent. For all other classes texture proved to be a useful feature for CBIR.

In general, for most classes an acceptable precision was achieved; for some queries even excellent (e.g., see Figures B.12 and B.13 in Appendix B). However, the performance differed considerably between the classes and between the queries within these classes.

Table 12.3: The average number of images selected (i.e., indicating the precision) and the standard deviation (between brackets), for both engines (color and color&texture) on both feature and semantic precision.

Class	Color-based		Color&Texture-based	
	Feature	Semantic	Feature	Semantic
dinos	10.14 (5.04)	8.90 (4.99)	11.89 (4.11)	11.30 (4.54)
flowers	7.14 (3.92)	4.75 (2.12)	7.05 (5.08)	4.05 (2.11)
food	6.81 (3.11)	2.20 (2.14)	5.56 (4.57)	2.85 (3.36)
women	6.31 (4.16)	5.20 (2.98)	8.40 (5.24)	5.60 (2.64)
waterfall	11.27 (2.64)	7.05 (1.76)	11.46 (2.75)	7.90 (2.22)
cats	6.10 (4.03)	8.10 (3.39)	8.80 (4.94)	8.85 (3.62)
dogs	5.66 (2.54)	6.48 (2.50)	7.45 (5.06)	7.35 (2.57)

12.7 Discussion

The present chapter provided an overview of the development cycle of new object-based CBIR techniques. These were evaluated in a CBIR benchmark, which provided the Corel image database and an interface module, for the engines developed. In order to provide intuitive results for users based on computationally cheap generic techniques, we mimicked human visual processing characteristics, utilizing the 11 color categories, four texture features derived from the color correlogram, and image segmentation by agglomerative merging. A central region from the image was chosen, such that it had a high probability to represent the object, which is the subject of the image. With a feature vector of 15 elements (i.e., the 11 colors + 4 texture features) and a segmentation algorithm based on the 11 color categories, the techniques introduced are very cheap.

The final color&texture-based engine proved to have a good precision. However, the engine is not generic applicable since it needs to be fine-tuned for different classes of images. This is due to the different background scenes against which the images in the Corel image database are photographed. So, the amount to which the objects differ in texture from their background is variable. This variability in texture differences between classes is the reason the parameters have to be fine-tuned for each object class.

In Section 12.5.2, we discussed the difference between feature and semantic precision. This is of interest since often the claim is made that a CBIR engine retrieves images based on semantic properties, while actually retrieval is based on image features that correlate with semantic categories. Feature precision was significantly higher than semantic precision for both the color-based engine and the color&texture-based engine. These results indicate that, when the retrieval results were not semantically relevant, they were intuitive to the users. Especially, heterogeneous image classes proved to be a problem for semantic precision, which was illustrated by the class food. We do not expect that images of such classes can be adequately classified or retrieved from a database using an object-based approach.

This chapter describes the development of an efficient OBIR engine that provides good retrieval results. Its efficiency is founded on principles inspired by human perception. Moreover, it provides intuitive results for its users. Hence, an important step is made toward bridging the semantic gap present in CBIR.

In this chapter, OBIR was conducted. However, the objects' shape could not be utilized, since only coarse segments were extracted. Subsequently, matching was performed by color and texture features. In the next chapter, a shape extraction method is introduced. In addition, shape-matching was conducted and OBIR will be done, using color, texture and shape features. Moreover, a solution concerning the problem of parameter tuning will be presented.

13

Human-centered object-based image retrieval

Abstract

A new object-based image retrieval (OBIR) scheme is introduced. The images are analyzed using the recently developed, human-based 11 colors quantization scheme and the color correlogram. Their output served as input for the image segmentation algorithm: agglomerative merging, which is extended to color images. From the resulting coarse segments, boundaries are extracted by pixelwise classification, which are smoothed by erosion and dilation operators. The resulting features of the extracted shapes, completed the data for a <color, texture, shape>-vector. Combined with the intersection distance measure, this vector is used for OBIR, as are its components. Although shape matching by itself provides good results, the complete vector outperforms its components, with up to 80% precision. Hence, a unique, excellently performing, fast, on human perception based, OBIR scheme is achieved.

This chapter is an adapted version of:

Broek, E. L. van den, Rikxoort, E. M. van, and Schouten, Th. E. (2005). Human-centered object-based image retrieval, *Lecture Notes in Computer Science (Advances in Pattern Recognition), 3687*, 492–501.

13.1 Introduction

More and more, the world wide web (www), databases, and private collections are searched for audio, video, and image material. As a consequence, there is a pressing need for efficient, user-friendly, multimedia retrieval and indexing techniques. However, where speech and handwriting recognition algorithms are generally applicable, image and video retrieval systems are only successful in a closed domain. These techniques have in common they are computational expensive and their results are judged as non-intuitive by its users.

In this chapter, these drawbacks are tackled, for the field to content-based image retrieval (CBIR). An object-based approach on CBIR is employed: object-based image retrieval (OBIR), inspired by the findings of Schomaker, Vuurpijl, and De Leau [249], who showed that 72% of the people are interested in objects when searching images. Moreover, a human-centered approach is chosen, based on the 11 color categories used by humans in color processing, as described in Chapter 2-5. These 11 color categories are also utilized for texture analysis, as discussed in Chapter 10, and for image segmentation, done by agglomerative merging (see Section 12.2.1). From the resulting, coarse image segments, the shape of the object is derived using pixelwise classification (Section 13.2). Next, erosion and dilation operations are applied on the boundary in order to smooth it. Section 13.3 introduces the shape matching algorithm. OBIR is conducted using four query schemes (see Section 13.4): two of them are based on color and texture, one on the object boundaries, and one on their combination. The results are presented in Section 13.5 followed by a discussion in Section 13.6.

13.2 Feature extraction

As shown in the previous chapters, color and texture are important features in image recognition for both humans and computers. Moreover, the parallel-sequential texture analysis, as introduced in Chapter 10, illustrated the complementary characteristics of global color analysis and color induced texture analysis. Therefore, parallel-sequential texture analysis was also applied in the current study. It utilizes the 11 color categories and a combination of four texture features derived from the color correlogram, as introduced in Chapters 9.

Shape extraction was conducted in three phases: (i) coarse image segmentation, as was introduced in the previous chapter (ii) pixelwise classification based on the 11 color categories, and (iii) smoothing. The coarse image segmentation uses only texture information to segment the image in texture regions. In the pixelwise classification phase, only color information is used because the regions are too small for our texture descriptor to be informative.

The coarse image segmentation is done by agglomerative merging. For the current

dataset, the stopping criterion Y for merging is determined to be 0.700. When the coarse segmentation phase is complete, the center segment of the image is selected to be the object of interest for OBIR.

After the center object has been identified in the coarse segmentation phase, pixelwise classification [201] is applied to improve localization of the boundaries of the object. In pixelwise classification, each pixel on the boundary of the center object is examined. A disk with radius r is placed over the pixel and the 11 color histogram is calculated for this disk and all adjacent segments. Next, the distance between the disk and the adjacent segments is calculated, using the intersection distance measure [31] based on the 11 color histogram. The pixel is relabeled if the label of the nearest segment is different from the current label of the pixel. This process, visualized in Figure 13.1, is repeated as long as there are pixels that are being relabeled.

The radius r of the disk determines how smooth the resulting boundaries are: a small radius will produce ragged regions, a larger radius will produce smoother boundaries but may fail in locating the boundaries accurately. In order to tackle these problems we used a two-step approach: In the first iterations, a relatively small radius of 5 is used, in order to locate the boundaries correctly. Secondly, a radius of 11 is used to produce more stable segments.

Although the pixelwise classification phase produces correct object boundaries, the shapes are smoothed to optimize for the shape matching phase. Smoothing is done using two fundamental operations: dilation and erosion, as were introduced in Section 4.2 of Chapter 4 The smoothing is done by an opening operation; i.e., an erosion followed by a dilation using the same structuring element for both operations. The latter was done with a square marker (B) of size 5×5 pixels. Hence, the process of shape extraction is completed. The complete process of shape extraction is illustrated in Figure 13.2.

13.3 Shape matching

Shape matching has been approached in various ways. A few of the frequently applied techniques are: tree pruning, the generalized Hough transform, geometric hashing, the alignment method, various statistics, deformable templates, relaxation labeling, Fourier and wavelet transforms, curvature scale space, and classifiers such as neural networks [3].

Recently, Andreou and Sgouros [3] discussed their: "turning function difference", as a part of their G Computer Vision library. It is an efficient and effective shape matching method. However, Schomaker et al. [249] introduced a similar approach five years before. In the current research, the latter, original approach is adopted. This "outline pattern recognition", as the authors call it, is based on three feature vectors containing: (i) x and y co-

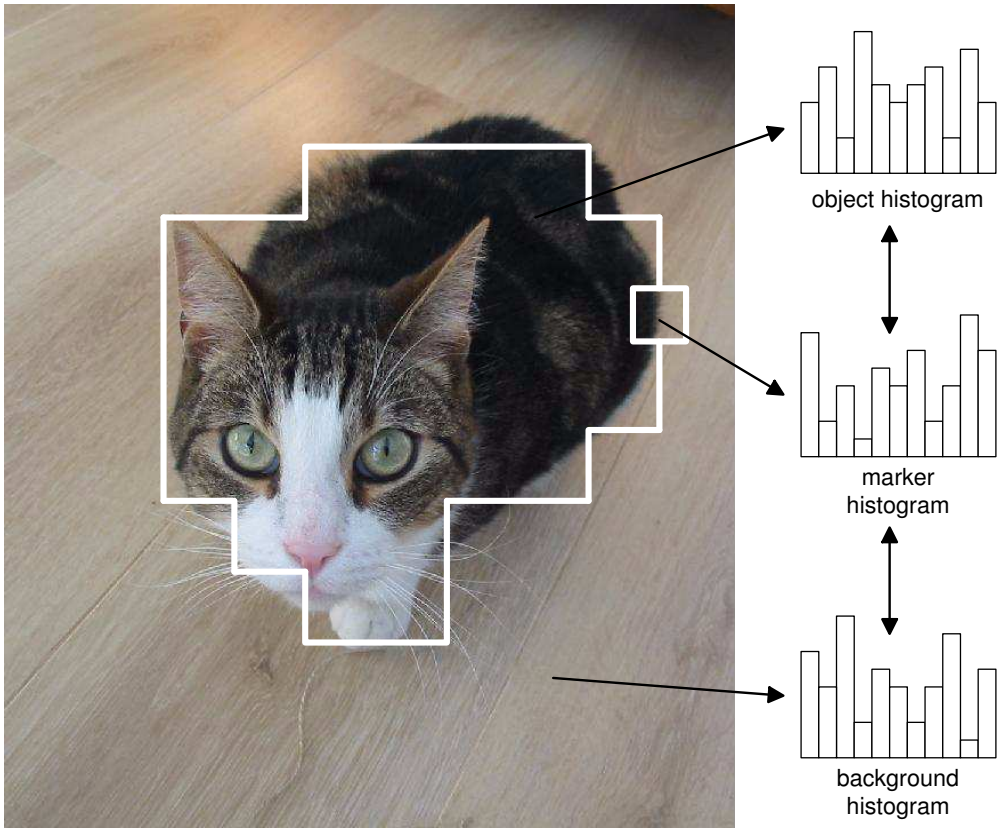


Figure 13.1: The process of pixelwise classification illustrated. A pixel at the boundary is selected and a marker is placed over it. Next, the color histogram over this marker is calculated as well as the histograms of the center segment and the background. The histogram over the marker is compared to the other histograms and the pixel is assigned to the area with the most similar histogram (of the background or the object). A large, full color version of this figure is provided as Figure B.17 in Appendix B.

ordinates, normalized using the center of gravity of the shape and the standard deviation of the radii, for all points (x, y) (ii) the running angle (θ) along the edge of the segment $(cos(\theta), sin(\theta))$, which contains more information on the local changes of direction, and (iii) the histogram of angles in the shape: the probability distribution $p(\theta)$ [249].

The algorithm proved to be translation, scale, and rotation invariant. Based on this algorithm, the outline-based image retrieval system Vind(X) was developed and has been used successfully since then. Vind(X) relies on outline-outline matching: the user draws an outline, which is the query. This outline is matched against the outlines of objects on images, present in its database. Subsequently, the images containing the best matching outlines are retrieved and shown to the user.

The Vind(X) system provides excellent retrieval results. However, in order to make its techniques generally applicable, automatic shape extraction techniques had to be developed. Moreover, these techniques had to be computationally cheap in order to preserve its fast retrieval, as much as possible. The latter was already achieved by the techniques as

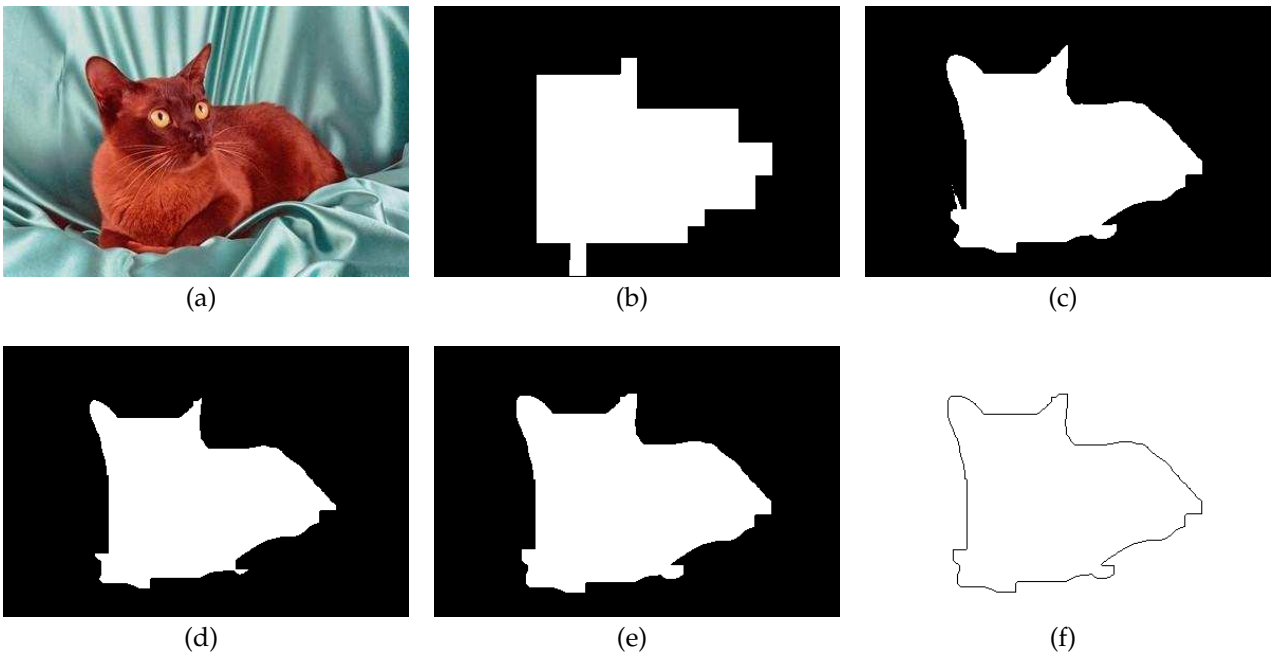


Figure 13.2: (a) The original image (b) The coarse segmentation (c) The object after pixelwise classification (d) The object after erosion (e) The object after dilation (f) The final shape. See Figure B.1 in Appendix B for large, color versions of these images.

described in the previous sections. In combination with the matching algorithm of Vind(X), unsupervised OBIR was applied.

13.4 Method

In Section 13.2, color, texture, and shape features are defined. They are combined and used in four distinct query schemes for object matching, using four vectors:

1. color and texture (parallel-sequential texture analysis), for object vs complete images
2. color and texture (parallel-sequential texture analysis)
3. shape
4. color and texture (parallel-sequential texture analysis) and shape combined

Feature-based and shape-based image retrieval was employed by two separate retrieval engines, connected to the same database, both using the intersection distance measure for ranking their results. For both engines, the number of retrieved images (n) could be chosen by the user. All query schemes performed an object - object comparison, except scheme 1 for which object features are matched with the features of the complete images in the database. For query scheme 4, for each image its ranks on both engines are summed and divided by two.

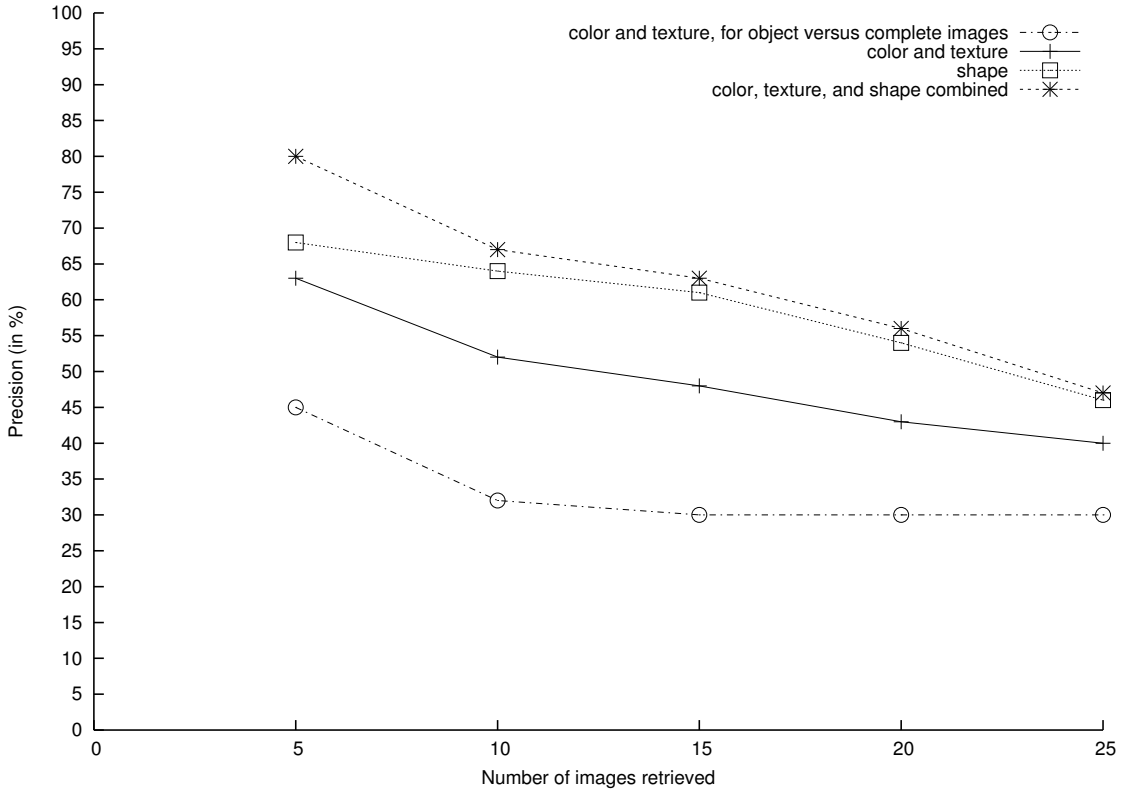
Figure 13.3: Sample images from the database used. See Figure B.19 in Appendix B for large, color versions of these six sample images.

In total, the database used, consists of 1000 images gathered from the Corel image database, a reference database for CBIR applications, and from the collection of Fei-Fei [84]. Since we are interested in objects, the six categories chosen represent objects: cats, leaves, revolvers, motorbikes, pyramids, and dinosaurs.

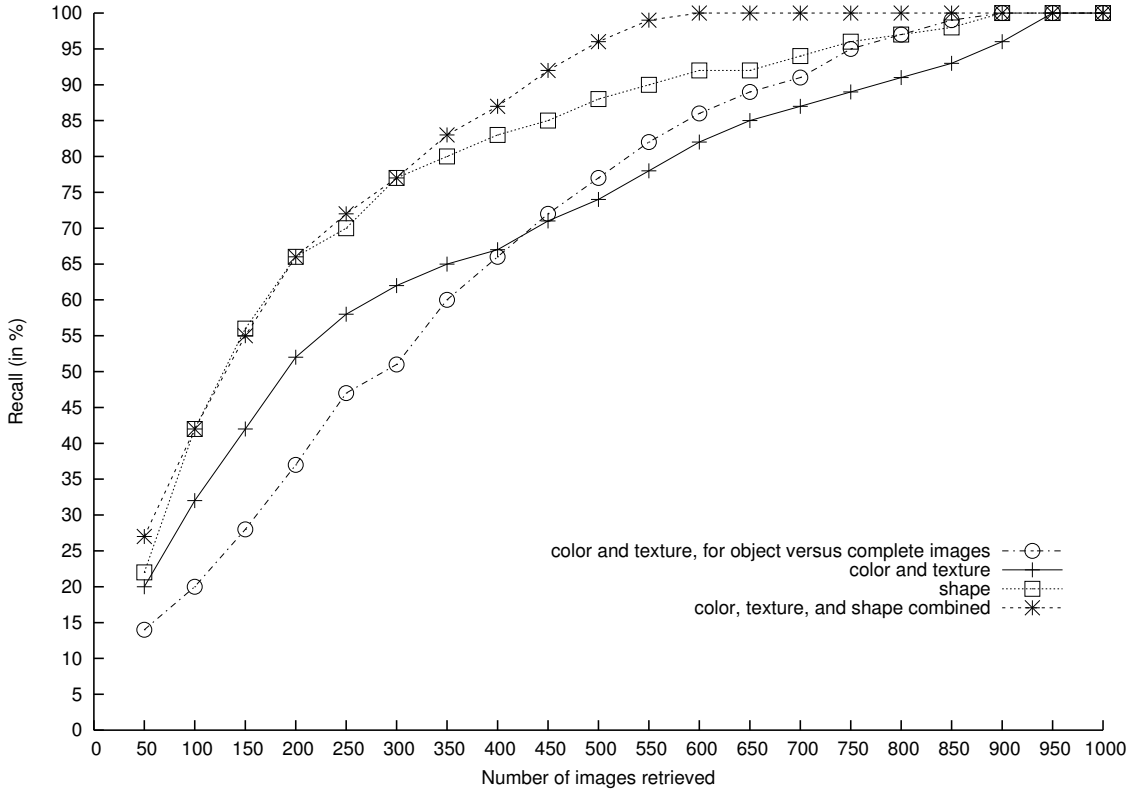
Adopted from the field of Information Retrieval, the performance of CBIR systems can be determined by the measures recall and precision. Recall signifies the proportion of relevant images retrieved from the database in response to the query. Precision is the proportion of retrieved images that is relevant to the query.

13.5 Retrieval results

Recall and precision are calculated for each of the four different query schemes, as defined in Section 13.4, using a variable number of images retrieved. The precision of the retrieval results for the four schemes are plotted in Figure 13.4(a), for 5–25 images retrieved. The recall of the retrieval results for the four schemes are plotted in Figure 13.4(b), for the complete dataset. All four schemes performed well, as shown in Figure 13.4(a) and 13.4(b). However, note that with the combined approach, four of the top five images are relevant; i.e., an average precision of 80% was achieved. Moreover, the recall achieved with the combined approach converges much faster to 100% than with the other approaches.



(a)



(b)

Figure 13.4: Average precision (a) and recall (b) of retrieval, with global and local color&texture features, outline of extracted objects from images, and their combination.

13.6 Discussion

The rationale of the CBIR approach presented in this chapter is that it be human centered. This is founded on two principles: (i) CBIR should be object-based and (ii) it should utilize the 11 color categories, as used by humans in color processing [31]. Both principles contribute to efficient CBIR, providing intuitive results for users. It was shown that the 11 color categories work well for describing color distributions (see Chapter 2), for the extraction of texture descriptors (see Chapter 9), and for object segmentation (see the previous chapter), as illustrated by the recall and precision of the retrieval results.

The success of matching the 2D shapes of segmented objects with each other is striking. This can, at least partly, be explained by the fact that "photographers generate a limited number of 'canonical views' on objects, according to perceptual and artistic rules" [249]. Moreover, even in the most recent research still (computationally expensive) gray-scale techniques are applied [260]. In contrast, we are able to extract shapes from color images. This is very important, since most of the image material available on the www and in databases is color.

In contrast with the reality on the www, the images in our database all contain images of objects against a rather uniform background, as illustrated in Figure 13.3. With our database, a first step is made toward processing real world images, where in comparable, recent work [91], object images are used that lack a background.

Despite the success of the current approach on real world images, it also has some drawbacks. First, it should be noted that the number of categories and its members were limited and follow-up research should be conducted with a larger database, incorporating a large number of categories. Second, in further developing the engine, the segmentation parameter should be set dynamically; i.e., setting the parameter to a minimum value and resetting it dynamically during the merging phase, based on the texture differences between the remaining blocks. This would obviate the current dependency on a good pre-defined parameter setting. Third, the ultimate goal would be to identify all objects in an image, instead of one, as is currently the case. Fourth, we expect that the use of artificial classifiers can improve the results, compared to the distance measures, used in the current research. When these drawbacks have been overcome, the resulting CBIR engine can be applied to real-world images instead of only to object-classes.

In this chapter, a highly efficient scheme for the extraction of color, texture, and shape features is introduced. Combined with the intersection distance measure, it forms the basis of a unique, good performing, fast object-based CBIR (OBIR) engine, which provides results intuitive for its users.

14

Epilogue

This chapter is partly based on fragments of:

Broek, E. L. van den, Vuurpijl, L. G., Kisters, P. M. F., and Schmid, J. C. M. von (2002). Content-Based Image Retrieval: Color-selection exploited. In M.-F. Moens, R. De Brusser, D. Hiemstra, and W. Kraaij (Eds.), *Proceedings of the 3rd Dutch-Belgium Information Retrieval Workshop*, p. 37-46. December 6, Belgium - Leuven.

Broek, E. L. van den, Vuurpijl, L. G., Hendriks, M. A., and Kok, T. (2003). Human-Centered Content-Based Image Retrieval. In G. Strube and R. Malaka (Eds.), *Program of the Interdisciplinary College 2003: Applications, Brains & Computers*, p. 39-40. March 7-14, Germany - Günne, Möhnesee.

Broek, E. L. van den, Kisters, P. M. F., and Vuurpijl, L. G. (2004). Design Guidelines for a Content-Based Image Retrieval Color-Selection Interface. In B. Eggen, G. van der Veer, and R. Willems (Eds.), *Proceedings of the Conference on Dutch directions in HCI*, ACM International Conference Proceeding Series. June 10, The Netherlands - Amsterdam.

As a result of the expanding amount of digital image material, the need for content-based image retrieval (CBIR) emerged. However, in developing CBIR-techniques seldomly the user and his characteristics were taken into account and, subsequently, limitations of mere technical solutions became apparent (see Chapter 1). Furthermore, the importance of these technical solutions is eminent since we have still a long way to travel before understanding man (i.e., the user). This thesis made an attempt to exploit knowledge concerning human color perception in image processing algorithms and, in parallel, improve the existing image processing algorithms. Hereby, the human was constantly taken into the loop.

We started with fundamental research toward human color processing (see Chapter 3). This resulted in a unique color space segmentation, driven by experimental data concerning the 11 color categories, known to be used by humans since half a century [40, 41]. For the computation of the color categories, the new Fast Exact Euclidean Distance (FEED) transform was introduced (see Chapter 4 and 5 and Appendix C), the fastest exact Euclidean distance transform available. This color space segmentation can function as a highly efficient, human-based, color quantization scheme, as was illustrated in Chapters 7 and 8.

With respect to texture analysis techniques, mostly color is ignored [35, 36]. Subsequently, problems can arise since two distinct colors can have the same intensity. When an image consisting such colors is converted to a gray-scale image, compounds of the image will merge (e.g., object and background can become one). After a comparison of several texture analysis techniques, we developed a new, parallel-sequential texture analysis approach, with up to 96% correct classification performance (see Chapter 10). Moreover, in the research "mimicking human texture classification", artificial and human texture classification were compared to each other (see Chapter 11), in a unique experimental setup.

Using the 11 color categories and the texture analysis scheme developed, coarse image segmentation was conducted (Chapter 12) and subsequently, exact shapes were extracted by pixelwise classification, followed by smoothing operators (Chapter 13). The shapes extracted were analyzed using the Vind(X) engine. With that, for all three features (i.e., color, texture, and shape), human-based techniques have been developed to extract them from unannotated image material. Using these techniques, object-based image retrieval was conducted (see Chapters 12 and 13), with excellent results.

During several phases of the research, the feature extraction techniques were tested in a newly developed, online CBIR benchmark (Chapters 6, 7, and 8). Object-based image retrieval, exploiting color, texture, and shape features, resulted in a high retrieval precision of up to 80% (Chapter 13). Hence, a human-based, computationally very efficient, object-based image retrieval engine was launched.

Each chapter was equipped with its own discussion concerning the research presented in that chapter. Therefore, this epilogue provides three general issues concerning CBIR and sketches follow-up research, before closing with a general conclusion. We discuss three

topics only touched or even ignored so far: (i) intelligent CBIR or "when and why CBIR", concerning the (dis)use of CBIR, (ii) issues concerning user interfaces (UIs) for CBIR systems; the research conducted with respect to this topic was omitted from this thesis for reasons of conciseness, and (iii) gray-scale image retrieval.

14.1 Intelligent Content-Based Image Retrieval

In the general introduction (Chapter 1), we have defined intelligent CBIR as CBIR that utilizes knowledge concerning human cognitive abilities. On the one hand, this should facilitate in providing intuitive results for users and, on the other hand, this should result in efficient image analysis schemes.

However, intelligent CBIR can also be approached from another point of view: the integration of Information Retrieval (IR) and CBIR techniques. From this moment on, we will denote the integration of IR and CBIR as iCBIR, where the 'i' is derived from intelligent.

Modern search engines like Google illustrate that "Classic IR methods are fast and reliable when images are well-named or annotated. [42]" However, "keywords have become a serious constraint in searching non-textual media. Search engines that provide facilities to search pictures (e.g., AltaVista and Google) usually link to specialized, closed, image databases. The results, however, are in no way parallel to the success of text retrieval [115]." Moreover, it is evident that they are incapable of searching in unannotated image collections. CBIR [235, 270] methods are capable of searching in such collections. [42]

14.1.1 Levels of image queries

Three levels of abstraction can be distinguished with image queries [117]:

1. Primitive features; i.e., color, texture, and shape
2. Derived features:
   (a) type of objects
   (b) individual objects / persons
3. Abstract attributes:
   (a) names of events or types of activity
   (b) emotional or religious significance

The higher the level of abstraction, the more problems CBIR systems will encounter in satisfying the user needs.

In general, CBIR techniques excel in deriving primitive features from image material. However, since not all techniques applied are intuitive for humans, the results achieved with them are not either. Nevertheless, most CBIR systems fully rely on image retrieval using primitive features.

Types or classes of objects are defined as such because they share common characteristics. Often these characteristics can be pinpointed, using primitive features. However, more often problems emerge in defining the features of a prototype of a certain type (e.g., Ferrari, politicians). More general types of objects, can be defined by prototypes; e.g., cars, humans. In contrast, more specific types of objects (e.g., Ferrari F40, Balkenende) are impossible to describe using primitive features. Only up to a restricted level of complexity object search can be done; e.g., face recognition [13, 17, 152, 212]. In general, with such queries one still relies on text-based methods. For example, when searching for photos of particular objects (e.g., the "Kronenburger Park, Nijmegen, The Netherlands") by keywords, or to search for photos of a particular class of objects (e.g., vegetables), by browsing catalogs. In contrast, with general object or scene queries (e.g., when searching photos of "sunsets", "landscapes", and "red cars") one can conveniently rely on CBIR methods.

The highest level of abstraction is found with names of events or types of activity and with emotional or religious significance. It is easily imaginable that such categories of photos are not suitable for CBIR methods. For instance, impressionist or abstract paintings are hard to classify. More important than color, texture, and shape characteristics of the painting, is a painting's expression and how it is experienced by its viewers. For now, such a description is far out of the reach of CBIR techniques.

What can be done using primitive image features is deriving the style of paintings (e.g., the period in which they were made) [33, 34, 111], determine the painter who made them [33, 34, 111], and verify whether or not they are original works [74, 164, 282]. Then, based on these characteristics of the content of art objects, these objects can be classified. These classifications can even be connected to emotional expressions accompanying the classes of paintings. Using relevance feedback and intelligent (learning) algorithms (e.g., artificial neural networks), such classes can be annotated.

In general, on the one hand, one can state that text-based image retrieval methods can be used to overcome the limitations of CBIR methods. On the other hand, CBIR methods can assist human annotators in their task of annotating image databases. For example, first, automatic catalogs of image databases can be generated, using CBIR methods; second, users can refine the coarse classification made by the CBIR methods.

So far, we held out a prospect of the combination of both CBIR and text-based image retrieval. This is in line with the suggestions of Lai, Chang, Chang, Cheng, and Crandell [155]: "For most users, articulating a content-based query using these low-level features can be non-intuitive and difficult. Many users prefer to using keywords to conduct

searches. We believe that a keyword- and content-based combined approach can benefit from the strengths of these two paradigms." Currently, the NWO ToKeN-project VindIT employs the exploration of this combination [323]. The VindIT project aims to extract additional information coming forth from the combination of text and content-based features, instead of merely combining both features. At the end of the 90s, the first results of such an approach already yielded promising results [49]. Hence, with a successful combination of IR and CBIR, truly intelligent CBIR can be conducted. However, an even higher increase in performance may be gained by facilitating a proper human-computer interaction. Since the intermediate between user and system is the user interface (UI), these UIs should be a topic of research on their own, within the field of CBIR.

14.2 CBIR User Interfaces (UIs)

A CBIR system can be described by three compounds: (i) a query definition interface, (ii) a search engine (and database), and (iii) the presentation of the retrieval results. For both the first component and the third component, a proper UI is of the utmost importance. Van den Broek, Vuurpijl, Kisters, and Von Schmid [42] were the first "to conduct a review on ten online CBIR engines, emphasizing interface aspects and judging human-computer interaction." Two years later, Van den Broek, Kisters, and Vuurpijl [29] still had to conclude that "no extensive review on user-interfaces of CBIR systems is present today." Existing CBIR reviews, such as that of Gevers and Smeulders [90], Venters and Cooper [305], Veltman and Tanase [303], and Veltkamp, Burkhardt, and Kriegel [302], emphasize the various image retrieval techniques, but not their interfaces. Others, such as Steiner [283], only briefly discuss the usability of 36 freely available web based color selectors, in a non-CBIR setting. Thus, we may conclude that the role of UIs in CBIR is underexposed. However, the UIs are the interfaces between the CBIR engine and its users and should fit the users needs. The next subsections describe all components of the UI needed to define a query and of the UI that present the results.

14.2.1 CBIR color selectors

For specifying the color of a CBIR query, a color selector is needed. Most CBIR color selectors evolved from copies of interfaces in graphics applications. Color selectors in the graphics industry were present, years before the first CBIR engine was born. However, color selectors for the graphics industry do have other demands than those for CBIR [42]; e.g., subtle level crossings do not have to be made for CBIR, but are custom in graphics design. [29]

Van den Broek, Kisters, and Vuurpijl [29] mentioned three considerations in the design of a CBIR color-selection UIs:

1. Human color memory is poor, it stores color in only 11 categories.
2. The more colors are present, the harder the selection is: both from perceptual and motor point of view.
3. Color-selectors for graphic design and CBIR systems cannot be interchanged.

Based on these considerations, a prototype CBIR color selector was developed [29, 42]. It was combined with a sketchpad for shape-based image retrieval [249], taken from the Vind(x) system [308] (see 14.2.3).

14.2.2 Defining texture

Color can be selected from a color selector. How to design such a selector is far from trivial but is possible, as is shown in many applications available today. However, an even more challenging issue is how users should define texture. As Celebi and Alpkoçak [50] already noted: "In forming an expressive query for texture, it is quite unrealistic to expect the user to draw the texture (s)he wants."

Two alternatives are possible for drawing texture: A palette of textures can be used, which facilitates texture-by-example querying or the option to textually describe texture can be provided. Perhaps this would be possible when using a set of restricted keywords with which textures can be described. For the latter purpose, the three main characteristics (i.e., repetivity, contrast, and coarseness) as denoted by Rao and Lohse [218] could be used. However, the descriptions and their interpretation would be subject to subjective judgments and the limitations of human cognition. So, texture description by text is hard, if possible at all. This leaves one UI that is feasible for defining texture: the palette of textures.

14.2.3 Sketching

Shape definition by sketching is used in the Vind(X) system and demonstrated to be useful. However, drawing with a mouse is very hard. Making drawings by use of pen and tablet is easier but, for untrained users, still very hard. Moreover, the query-by-sketch paradigm is not used outside a limited database. So, is this paradigm useful in a less restricted domain?

As shown in Vuurpijl, Schomaker, and Van den Broek [308], the quality of drawings, and with that their usability, differs substantially. Moreover, most users are not equipped with sufficient drawing techniques to draw canonical views of images. Figure 14.1 presents drawings of users as collected and presented in [308], which are all drawings of objects (i.e., humans, horse, table, and tree) as seen from their front or side. Since most photographers take photos of objects from a canonical view [18, 246, 249, 291], this limits the mapping between segmented shapes from photos and the sketches as provided by users.

To summarize, the use of color for query-by-memory seems feasible. Texture can probably not be employed in query-by-memory settings. Sketch (or shape) based retrieval can be performed. However, its use has not been demonstrated on a large database, with various types of images.

14.2.4 Shape and color

In the field of CBIR, the relation between color and shape has received little attention [42]; this is no different for the current thesis. However, this thesis has shown that this relation is present and is of importance. In Chapters 9–11, it was proved that color (C) defines texture (T); subsequently, in Chapters 12–13, texture (T) on its turn is exploited for image segmentation purposes and for shape (S) extraction. Hence, $C \rightarrow T$, $T \rightarrow S$ implies $C \rightarrow S$, where $\rightarrow$ is defined as: "is used to extract". This section discusses this direct relation between color and shape.

Color and shape features were combined in the design of a query-by-memory CBIR interface (see [42]; this research was not included in this thesis). This combination was based on findings, which state that shape and color influence each other. Their relation is twofold: (i) color influences human object recognition and (ii) the 'shape category' of an object may influence the perceived color of it. The influence of color perception on object recognition is described by Goldstone [93]. In his article "Effects of categorization on color perception", he states that: "high-level cognitive processes do not simply operate on fixed perceptual inputs; high level processes may also create lower level percepts". Seen from this perspective, it can be stated that $C \Leftrightarrow T$, $T \Leftrightarrow S$ and, subsequently, $C \Leftrightarrow S$, where $\Leftrightarrow$ is defined as: "influence each other".

The relation between shape and perceived color is also observed by Sacks [239] who describes the horror people experience when perceiving objects, after they lost their ability to see color. Meadow [180] described the disabilities his patients had in distinguishing between objects, due to the loss of the ability of seeing color. He further notes that these problems are especially important for the recognition of those objects that rely on color as a distinguishing mark (cf. an orange and a grapefruit). So, in both the design of CBIR engines as well as for the CBIR query interfaces, this relation should be taken into account.

In order to conduct research toward (CBIR) UI design, recently an experimental environment was developed in which UIs can be tested. This provides the means for recording all user behavior for multiple user interfaces and stimuli, within an experimentally controlled design. Currently, two large scale experiments are prepared in which five color selection UIs will be tested (Color-Selection User Interface Testing (C-SUIT)), part of the project Scientific User Interface Testing (SUIT) [32]. In time, these experiments can be expected to result in guidelines for (CBIR) UIs.

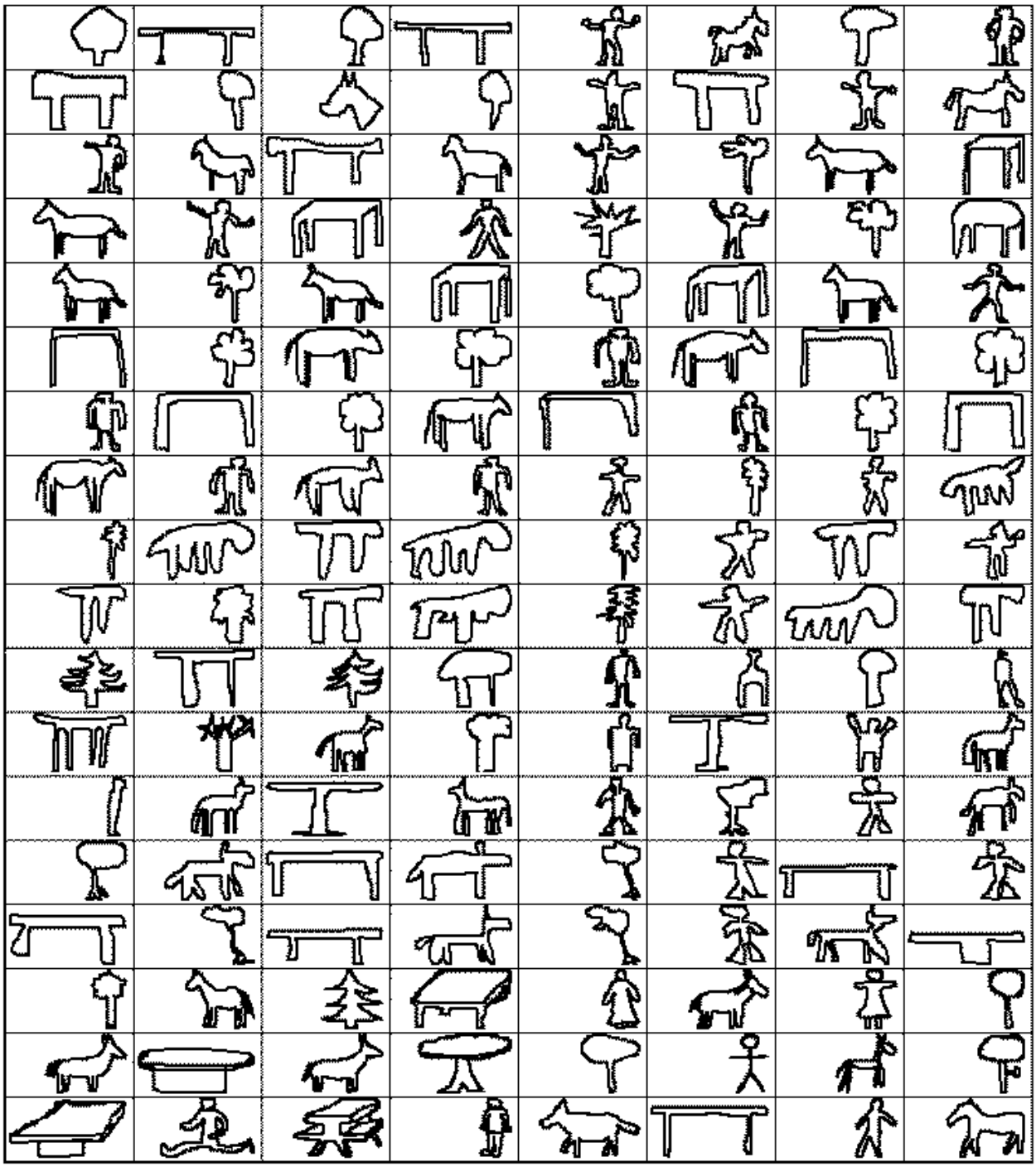


Figure 14.1: A selection of sketches drawn, as presented in Vuurpijl, Schomaker, and Van den Broek [308]. Participants were asked to draw humans, horses, tables, and trees. Note that most of these sketches provide the frontal or side view of the objects.

14.2.5 Presentation of CBIR results

Not only for query definition purposes but also for the presentation of the CBIR retrieval results an UI is needed. This is of the utmost importance since people visually browse through these results [27]. However, until now there is a lack of fundamental research concerning the presentation of the results of a CBIR query. De Greef and Van Eijk [72] are an exception with their research toward "Visually searching a large image database". They found that "manual browsing was more effective but less efficient than computer controlled browsing. [27]" "A grid of 9 images per screen was found optimal for visualizing an image collection. In addition, they found that a difference in size between the (real-size) image (e.g., a painting) that was seen and its thumbnail version, as it is available in the image database, does not hinder recognition. [27]

Please note that these findings were fully utilized within the CBIR benchmark, as introduced in Chapter 6. With the design of a UI that presents the query results of the CBIR engine (running within the framework of the benchmark), the advice of Montfort et al. [72] was respected. We choose for manual browsing, used scaled images, and presented only 15 retrieved images per screen.

14.3 What to do without color? Gray-scale image retrieval

This research pursues CBIR methods for color images. However, a second category exists, that of gray-scale images. Although in this thesis only briefly touched for texture analysis, this category of images is also worth our attention. Gray-scale images are used in a range of professional settings; e.g., medical applications (see Section 1.3.2.B in Chapter 1), fingerprints [152, 174], optical character recognition (OCR), and handwriting recognition [195, 250].

Two types of gray-scale images should be distinguished: (i) Original black-and-white photos and (ii) Color images that are transformed to gray-scale images. An example of the first type is the Early Photography 1893–1960 collection [293] of The Rijksmuseum, the Leiden Print Room, and other Dutch, public collections. The second type occurs, for example, to reduce publishing costs (e.g., as is done for this thesis).

In order to determine the amount of lost information (if any) with the transform, Wang and Bovik [311] even introduced a numerical measure for gray-scale images: the "universal image quality index". Recently, Wang, Bovik, Sheikh, and Simonelli [312] generalized the measure to the "structural similarity index".

Next to old black-and-white images, up till now, professional photographers make

black-and-white images. They usually apply filters on gray (or brown) level photos to enhance their contrast; most used are orange and red filters (e.g., they can produce dramatically darkened skies). Subsequently, their quantized gray level representation is adapted. This emphasizes that, when applying CBIR methods on collections of gray-scale images, this issue should not be ignored. Hence, specialized gray level CBIR methods should be developed for this category of images. This was already proposed by Niblack et al. [194] in their "Updates to the QBIC system". However, their paper does not discuss the details of their work.

14.4 Future work

The CBIR techniques, as presented in this thesis, should be plugged into a general framework from which an online CBIR system can be developed. The benchmark, as introduced in Chapter 6, provides an excellent foundation for the envisioned CBIR system. It should be extended with a query definition interface, both for query-by-memory and query-by-example. In addition, its engine should be adapted so that it can incorporate several image processing schemes, which can be combined in any way preferred. Moreover, gray-scale image processing schemes should be developed and included in the CBIR system.

In a second phase of research, IR techniques should be incorporated. The combined force of IR and CBIR techniques would make the system usable in large and possibly even unrestricted domains. Only such a system can, in potential, be exploited in a commercial setting for image retrieval on the WWW. A CBIR system solely can be used in several professional settings as well as for managing photo-albums of customers.

The research as done within the project SUIT [32] will (in time) provide the guidelines to improve the UI for both the definition of the query as for the presentation of its results. However, the CBIR system can be further adapted to the individual user, by making use of relevance feedback [307]. Already in the 90s, a few researchers suggested to use relevance feedback in CBIR [63, 237, 280]. In the last five years, this advice was reinforced by multiple researchers [56, 92, 145, 154, 171, 326]. As was shown in the present research, users indeed vary considerably in their judgments of retrieval results; hence, relevance feedback could indeed increase user satisfaction substantially.

Next to images, video material can be retrieved based on its content [4, 121, 125, 126, 192, 211, 277, 302, 304]. For this purpose, frequently, so called key frames are selected, which can be analyzed as images. Henceforth, CBIR techniques can be utilized. Note that where with CBIR only one image is present, in content-based video retrieval a set of images describes a part of the video. In addition, note that, even more than with CBIR, content-based video retrieval suffers from time, computational, and storage (or space) complexity.

Therefore, the techniques presented in this thesis may be considered, since they provide a computational inexpensive and effective alternative.

CBIR and IR techniques can also be applied in a multimedia context. Then, the computational load of the processing schemes is even more important. As already described in 1994–1996 [108, 109, 196, 247, 248], true multimedia systems should, for example, also incorporate speech recognition as well as other speech analysis techniques [25, 203, 238]. Moreover, advanced schemes for human-computer interaction are needed to facilitate the communication between user and system [26, 203, 238, 307].

Frequently, agent technology is suggested for advanced human-computer interaction [24, 26, 134]. Advanced communication is envisioned between (software) agents and between users and agents, where agents represent their owner [76, 77, 132]. Such an agent needs a knowledge representation (e.g., an ontology) of the user's needs [264]; schemes that facilitate relevance feedback can be considered as such knowledge representation.

Already in 1996, the Profile (A Proactive Information Filter) project [265, 266] started, which envisioned an active information filtering system for Internet may be viewed as consisting of 'intelligent agents' which proactively, and usually autonomously, will serve the user. Over the years, the complexity of this aim came apparent. For example, to satisfy this aim, users have to trust their (software) agents [131, 170]. This can only be accomplished when these agents have a high enough level of autonomy and are able to reason by assumption [133]. When all these premises are satisfied then and only then, a first step is made toward true artificial vision and subsequently, intelligent CBIR, as meant in Chapter 1.

14.5 Conclusions

Most CBIR research focuses on the utilization of advanced (computationally expensive) algorithms. An important constraint for CBIR is the complexity of the algorithm chosen. The principles based on which humans process images are mostly ignored. On the one hand, this thesis did discuss and improve algorithms. On the other hand, human cognition was its foundation. Moreover, since humans are the users of CBIR systems and with that judge them, their (dis)abilities should be taken into account [117].

Each of the three features used for CBIR (i.e., color, texture, and shape), were developed from a human-centered perspective, where in parallel improvements to algorithms were an issue. The 11 color categories, as used by humans in processing color, function as the fundament for color analysis but also for texture and shape analysis. The texture analysis method developed, utilizes two intuitive features: the color histogram and the color correlogram. In addition, human and artificial texture classification were compared experimentally. The coarse segmentation and the subsequent shape extraction were founded on

the 11 color categories and the intuitive texture descriptors.

The utilization of color, texture, and shape enabled us to perform object-based image retrieval. This brings us to the concept on which the Vind(X) system was founded, which utilizes outline-outline matching. Where Vind(X) fully relied on its database of manually annotated outlines, the techniques introduced in this thesis provide the means to extract these automatically.

So, all ingredients for a human-centered, computationally efficient CBIR engine have been developed. Moreover, a newly developed online CBIR benchmark was introduced, which provides the means for the evaluation of CBIR techniques by its users. Most important, a new style of research was introduced, which integrates fundamental research, the development of algorithms, thorough evaluations, always taking the human in the loop, with the world as research laboratory. This approach yields both new, efficient image processing and CBIR techniques and can be considered an important step in demystifying human perception; consequently, an important step has been taken in bridging the semantic gap in CBIR.

Bibliography

[1] H. Abe, H. MacMahon, R. Engelmann, Q. Li, J. Shiraishi, S. Katsuragawa, M. Aoyama, T. Ishida, K. Ashizawa, C. E. Metz, and K. Doi. Computer-aided diagnosis in chest radiography: Results of large-scale observer tests at the 1996-2001 RSNA scientific assemblies. *RadioGraphics*, 23(1), 2003.

[2] A. Agresti and B.A. Coull. Approximate is better than exact for interval estimation of binomial proportions. *The American Statistician*, 52:119–126, 1998.

[3] I. Andreou and N. M. Sgouros. Computing, explaining and visualizing shape similarity in content-based image retrieval. *Information Processing & Management*, 41(5):1121–1139, 2005.

[4] S. Antani, R. Kasturi, and R. Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965, 2002.

[5] S. Antani, L. R. Long, and G. R. Thoma. A biomedical information system for combined content-based retrieval of spine x-ray images and associated text information. In *Proceedings of the 3rd Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2002)*, Ahamdabad, India, 2002.

[6] L. H. Armitage and P. G. B. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23(4):287–299, 1997.

[7] F. Aurenhammer and R. Klein. *Voronoi Diagrams*, chapter 5, pages 201–290. Amsterdam, The Netherlands: North-Holland, 2000.

[8] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1982.

[9] S. Battiato, G. Gallo, and S. Nicotra. Perceptive visual texture classification and retrieval. In M. Ferretti and M. G. Albanesi, editors, *Proceedings of the 12th International Conference on Image Analysis and Processing*, pages 524–529, Mantova, Italy, September, 17-19 2003.

[10] E. Benson. Different shades of perception: A new study shows how learning - and possibly language–can influence color perception. *APA Monitor*, 33(11):28, 2002.

[11] S. Beretti, A. Del Bimbo, and P. Pala. Content-based retrieval of 3d cellular structures. In *Proceedings of the Second International Conference on Multimedia and Exposition (ICME2001)*, pages 1096–1099, Tokyo, Japan, 2001. IEEE Computer Society.

[12] M. de Berg, M. van Kreveld, M. Overmans, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Berlin: Springer-Verlag, 2 edition, 2000.

[13] N. H. Bergboer, E. O. Postma, and H. J. van den Herik. Context-enhanced object detection in natural images. In T. Heskes, P. Lucas, L. Vuurpijl, and W. Wiegerinck, editors, *Proceedings of the 15th Belgium-Netherlands Artificial Intelligence Conference*, pages 27–34. Nijmegen: SNN, Radboud University Nijmegen, October 2003.

[14] J. R. Bergen and E. H. Adelson. Early vision and texture perception. *Nature*, 333(6171):363–364, 1988.

[15] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, Inc., San Jose, CA, 2002.

[16] B. Berlin and P. Kay. *Basic color terms: Their universals and evolution*. Berkeley: University of California Press, 1969.

[17] V. Blanz, A. J. O'Toole, T. Vettre, and H. A. Wild. On the other side of the mean: The perception of dissimilarity in human faces. *Perception*, 29(8):885–891, 2000.

[18] V. Blanz, M. J. Tarr, and H. H. Bülthoff. What object attributes determine canonical views? *Perception*, 28(5):575–599, 1999.

[19] R. van den Boomgaard, L. Dorst, L. S. Makram-Ebeid, and J. Schavemaker. Quadratic structuring functions in mathematical morphology. In P. Maragos, R. W. Schafer, and M. A. Butt, editors, *Proceedings of the International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, pages 147–154. Kluwer Academic Publishers, 1996.

[20] R. van den Boomgaard and H. J. A. M. Heijmans. Morphological scale-space operators: An algebraic framework. In J. Goutsias, L. Vincent, and D. S. Bloomberg, editors, *Proceedings of the International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, pages 283–290. Springer, 2000.

[21] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing: An International Journal*, 34:344–371, 1986.

[22] A.C. Bovik, M. Clarke, and W.S. Geisler. Multichannel texture analysis using localized

spatial filters. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 12:55–73, 1990.

[23] R. M. Boynton and C. X. Olson. Locating basic colors in the OSA space. *Color Research & Application*, 12:107–123, 1987.

[24] F. M. T. Brazier, C. M. Jonker, J. Treur, and N. J. E. Wijngaards. On the use of shared task models in knowledge acquistion, strategic user interaction and clarification agents. *International Journal of Human-Computer Studies*, 52(1):77–110, 2000.

[25] E. L. van den Broek. *Emotional Prosody Measurement (EPM): A voice-based evaluation method for psychological therapy effectiveness*, volume 103 of *Studies in Health Technology and Informatics*, pages 118–125. IOS Press - Amsterdam, The Netherlands, 2004.

[26] E. L. van den Broek. Empathic agent technology (eat). In L. Johnson, D. Richards, E. Sklar, and U. Wilensky, editors, *Proceedings of the AAMAS-05 Agent-Based Systems for Human Learning (ABSHL) workshop*, pages 59–67, Utrecht - The Netherlands, 2005.

[27] E. L. van den Broek and N. H. Bergboer. Eidetic: User-centered intelligent content-based image retrieval. In C. Klöditz, editor, *Proceedings of the third NWO ToKeN2000 symposium*, page 5, 2004.

[28] E. L. van den Broek, M. A. Hendriks, M. J. H. Puts, and L. G. Vuurpijl. Modeling human color categorization: Color discrimination and color memory. In T. Heskes, P. Lucas, L. Vuurpijl, and W. Wiegerinck, editors, *Proceedings of the 15th Belgium-Netherlands Artificial Intelligence Conference*, pages 59–68. SNN, Radboud University Nijmegen, 2003.

[29] E. L. van den Broek, P. M. F. Kisters, and L. G. Vuurpijl. Design guidelines for a content-based image retrieval color-selection interface. In B. Eggen, G. van der Veer, and R. Willems, editors, *Proceedings of the SIGCHI.NL Conference on Dutch directions in HCI*, ACM International Conference Proceeding Series, Amsterdam, The Netherlands, 2004. ACM Press - New York, NY, USA.

[30] E. L. van den Broek, P. M. F. Kisters, and L. G. Vuurpijl. The utilization of human color categorization for content-based image retrieval. *Proceedings of SPIE (Human Vision and Electronic Imaging IX)*, 5292:351–362, 2004.

[31] E. L. van den Broek, P. M. F. Kisters, and L. G. Vuurpijl. Content-based image retrieval benchmarking: Utilizing color categories and color distributions. *Journal of Imaging Science and Technology*, 49(3):293–301, 2005.

[32] E. L. van den Broek and T. Kok. Scientific user interface testing (SUIT), URL: http://eidetic.ai.ru.nl/thijs/C-SUIT/ [Last accessed on July 31, 2005].

[33] E. L. van den Broek, T. Kok, E. Hoenkamp, Th. E. Schouten, P. J. Petiet, and L. G.

Vuurpijl. Content-Based Art Retrieval (C-BAR). In *Proceedings of the XVIth International Conference of the Association for History and Computing*, 2005.

[34] E. L. van den Broek, T. Kok, Th. E. Schouten, and E. Hoenkamp. Multimedia for Art ReTrieval (M4ART). In *[submitted]*, 2006.

[35] E. L. van den Broek and E. M. van Rikxoort. Evaluation of color representation for texture analysis. In R. Verbrugge, N. Taatgen, and L. R. B. Schomaker, editors, *Proceedings of the 16th Belgium-Netherlands Artificial Intelligence Conference*, pages 35–42, Groningen - The Netherlands, 2004.

[36] E. L. van den Broek and E. M. van Rikxoort. Parallel-sequential texture analysis. *Lecture Notes in Computer Science (Advances in Pattern Recognition)*, 3687:532–541, 2005.

[37] E. L. van den Broek and E. M. van Rikxoort. Supplement: Complete results of the icapr2005 texture baselines. URL: http://www.few.vu.nl/~egon/publications/pdf/ICAPR2005-Supplement.pdf [Last accessed on July 31, 2005], 2005.

[38] E. L. van den Broek, E. M. van Rikxoort, and Th. E. Schouten. Human-centered object-based image retrieval. *Lecture Notes in Computer Science (Advances in Pattern Recognition)*, 3687:492–501, 2005.

[39] E. L. van den Broek, E. M. van Rikxoort, and Th. E. Schouten. An exploration in modeling human texture recognition. In *[submitted]*, 2006.

[40] E. L. van den Broek, Th. E. Schouten, and P. M. F. Kisters. Efficient color space segmentation based on human perception. [submitted].

[41] E. L. van den Broek, Th. E. Schouten, P. M. F. Kisters, and H. Kuppens. Weighted Distance Mapping (WDM). In N. Canagarajah, A. Chalmers, F. Deravi, S. Gibson, P. Hobson, M. Mirmehdi, and S. Marshall, editors, *Proceedings of the IEE International Conference on Visual Information Engineering (VIE2005)*, pages 157–164, Glasgow, United Kingdom, 2005. Wrightsons - Earls Barton, Northants, Great Britain.

[42] E. L. van den Broek, L. G. Vuurpijl, P. M. F. Kisters, and J. C. M. von Schmid. Content-based image retrieval: Color-selection exploited. In M.-F. Moens, R. de Busser, D. Hiemstra, and W. Kraaij, editors, *Proceedings of the 3rd Dutch-Belgium Information Retrieval Workshop*, volume 3, pages 37–46. University of Leuven, Belgium, December 2002.

[43] R. W. Brown and E. H. Lenneberg. A study in language and cognition. *Journal of Abnormal and Social Psychology*, 49(3):454–462, 1954.

[44] C. D. Burstein. Viewable with any browser campaign, URL: http://www.anybrowser.org/campaign/ [Last accessed on July 31, 2005].

[45] V. Bush. As we may think. *The American Monthly*, 176(1):101–108, July 1945.

[46] M. D. Byrne. *Cogntive architecture*, chapter 5, pages 97–117. Mahwah, NJ: Lawrence Erlbaum, 2002.

[47] W. Cai, D. D. Feng, and R. Fulton. Content-based retrieval of dynamic PET functional images. *IEEE Transactions on Information Technology in Biomedicine*, 4(2), 2000.

[48] S. K. Card, A. Newell, and T. P. Moran. *The Psychology of Human-Computer Interaction*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc., 1983.

[49] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, Santa Barbara, CA, June 1998. IEEE, IEEE. URL: http:/citeseer.nj.nec.com/lacascia98combining.html.

[50] E. Celebi and A. Alpkoçak. Clustering of texture features for content based image retrieval. In *Proceedings of the First International Conference on Advances in Information Systems*, volume 1909 of *Lecture Notes in Computer Sciences LNCS*. Springer-Verlag, 2000.

[51] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. Videoq: An automated content based video search system using visual cues. In *Proceeding of The Fifth ACM International Multimedia Conference*, pages 313–324, Seattle WA, November 1997. ACM Press.

[52] B. B. Chaudhuri, N. Sarkar, and P. Kundu. Improved fractal geometry based texture segmentation technique. *IEE Proceedings part E*, 140:233–241, 1993.

[53] H.D. Cheng, X.H. Jiang, Y. Sung, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.

[54] CHID Technical Coordinator. The Combined Health Information Database (CHID) online, URL: http://chid.nih.gov/ [Last accessed on July 31, 2005].

[55] G. Ciocca and R. Schettini. Using a relevance feedback mechanism to improve content-based image retrieval. In D. P. Huijsmans and A. W. M. Smeulders, editors, *Visual Information and Information Systems: Third International Conference (VISUAL'99)*, volume 1614 of *Lecture Notes in Computer Science*, pages 107–114. Springer-Verlag GmbH, 1999.

[56] G. Ciocca and R. Schettini. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition*, 34:103–199, 2001.

[57] A. Clark. *Being There: Putting Brain, Body, and World Together Again*. Cambridge: The MIT Press, 1997.

[58] E. Coiras, J. Santamaria, and C. Miravet. Hexadecagonal region growing. *Pattern Recognition Letters*, 19:1111–1117, 1998.

[59] Commission Internationale de l'Eclairage. International Commission on Illumination. URL: http://www.cie.co.at/ciecb/, [Last accessed on July 31, 2005].

[60] R. W. Conners, M. M. Trivedi, and C. A. Harlow. Segmentation of a high-resolution urban scene using texture operators. *Computer Vision, Graphics, and Image Processing*, 25:273–310, 1984.

[61] L. F. Costa and R. M. Cesar Jr. *Shape Analysis and Classification*. CRC Press, 2001.

[62] L. F. Costa, E. T. M. Manoel, F. Faucereau, J. van Pelt, and G. Ramakers. A shape analysis framework for neuromorphometry. *Network: Computation in Neural Systems*, 13(3):283–310, 2002.

[63] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos. Pichunter: Bayesian relevance feedback for image retrieval. In *Proceedings of International Conference on Pattern Recognition*, pages 361–369. Vienna, Austria, August 1996.

[64] G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:25–39, 1983.

[65] O. Cuisenaire and B. Macq. Fast and exact signed euclidean distance transformation with linear complexity. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP99)*, pages 3293–3296, March 1999.

[66] O. Cuisenaire and B. Macq. Fast euclidean transformation by propagation using multiple neighborhoods. *Computer Vision and Image Understanding*, 76(2):163–172, 1999.

[67] K. Czajka. Design of interactive and adaptive interfaces to exploit large media-based knowledge spaces in the domain of museums for fine arts. Master's thesis, University of Applied Science Darmstadt: Media System Design, 2002.

[68] G. Lu D. Zhang. Study and evaluation of different fourier methods for image retrieval. *Image and Vision Computing*, 23(1):33–49, 2005.

[69] P. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[70] N. Davenport. Council on library and information resources. URL: http://www.clir.org/, [Last accessed on July 31, 2004].

[71] I. Davies. A study of colour grouping in three languages: A test of the linguistic relativity hypothesis. *Britisch Journal of Psychology*, 89(3):433–453, 1998.

[72] H. P. de Greef and R. L. J. van Eijk. Visually searching a large image database: Manual browsing versus rapid visual presentation, [In preparation].

[73] T. C. J. de Wit and R. J. van Lier. Global visual completion of quasi-regular shapes. *Perception*, 31(5):969–984, 2002.

[74] A. I. Deac, J. C. A. van der Lubbe, and E. Backer. Optimal image generation for art databases. In V. Cappellini and J. Hemsley, editors, *Proceedings of Electronic Imaging and the Visual Arts (EVA) 2004*, pages 125–130, 2004.

[75] G. Derefeldt, T. Swartling, U. Berggrund, and P. Bodrogi. Cognitive color. *Color Research & Application*, 29(1):7–19, 2004.

[76] W. A. van Doesburg, A. Heuvelink, and E. L. van den Broek. TACOP: A cognitive agent for a naval training simulation environment. In M. Pechoucek, D. Steiner, and S. Thompson, editors, *Proceedings of the Industry Track of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-05)*, pages 34–41, Utrecht - The Netherlands, 2005. ACM.

[77] W. A. van Doesburg, A. Heuvelink, and E. L. van den Broek. Tacop: A cognitive agent for a naval training simulation environment. In F. Dignum, V. Dignum, S. Koendig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-05)*, volume 3, pages 1363–1364, Utrecht - The Netherlands, 2005. ACM.

[78] L. Dorst. *Objects in contact: boundary collisions as geometric wave propagation*, chapter 17, pages 349–370. Birkhäuser, 2001.

[79] S. Douglas and T. Kirkpatrick. Do color models really make a difference? In M. J. Tauber, V. Bellotti, R. Jeffries, J. D. Mackinlay, and J. Nielsen, editors, *Proceedings of CHI 96: Conference on Human Factors in Computing Systems*, pages 399–405. ACM, New York: ACM Press, 1996.

[80] A. Drimbarean and P. F. Whelan. Experiments in colour texture analysis. *Pattern Recognition Letters*, 22(10):1161–1167, 2001.

[81] D. B. Duncan. Multiple range and multiple F tests. *Biometrics*, 11(1):1–42, 1955.

[82] R. M. Evans. *An introduction to color*. New York: Wiley, 1948.

[83] O. D. Faugeras and W. K. Pratt. Decorrelation methods of texture feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):323–332, 1980.

[84] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In A. E. C. Pece, editor, *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, Washington, D.C., USA, June 27 – July 02 2004.

[85] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani,

J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC system. *IEEE Computer*, 28(9):23–32, 1995.

[86] D. Fontijne and L. Dorst. Modeling 3d euclidean geometry. *IEEE Computer Graphics and Applications*, (March/April):68–78, 2003.

[87] G. Frederix, G. Caenen, and E. J. Pauwels. Pariss: Panoramic, adaptive and reconfigurable interface for similarity search. In R. K. Ward, editor, *Proceedings of ICIP 2000 International Conference on Image Processing*, volume v.III, pages 222–225, 2000.

[88] S. Gauch, W. Li, and J. Gauch. The vision digital video library. *Information Processing & Management*, 33(4):413–426, april 1997.

[89] Th. Gevers and A. W. M. Smeulders. Color based object recognition. *Pattern Recognition*, 32(3):453–464, 1999.

[90] Th. Gevers and A. W. M. Smeulders. *Content-Based Image Retrieval: An Overview*. Prentice Hall PTR, 2004.

[91] Th. Gevers and H. M. G. Stokman. Robust histogram construction from color invariants for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):113–118, 2004.

[92] G. Giacinto and F. Roli. Bayesian relevance feedback for content-based image retrieval. *Pattern Recognition*, 37(7):1499–1508, 2004.

[93] R.L. Goldstone. Effects of categorization on color perception. *Psychological Science*, 5(6):298–304, 1995.

[94] Y. Gong. Advancing content-based image retrieval by exploiting image color and region features. *Multimedia Systems*, 7(6):449–457, 1999.

[95] R. C. Gonzales and R. E. Woods. *Digital image processing*. Prentice-Hall, Inc., New Jersey, 2nd edition, 2002.

[96] B. Göransson. *Usability design: A framework for designing usable interactive systems in practice*. PhD thesis, Uppsala University: Department of Information Technology, June 2001. ISSN 1404-3203.

[97] M. M. Gorkani and R. W. Picard. Texture orientation for sorting photos at a glance. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 459–464, 1994.

[98] D. Greenstein and S. E. Thorin. *The Digital Library: A Biography*. Digital Library Federation, Council on Library and Information Resources - Washington, D.C., 2002.

[99] W. Guan and S. Ma. A list-processing approach to compute Voronoi diagrams and

the Euclidean distance transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):757–761, 1998.

[100] V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *IEEE Computer*, 28(9):18–22, 1995.

[101] N. J. Gunther and G. Beretta. A benchmark for image retrieval using distributed systems over the internet: Birds-i. In *Proceedings of SPIE Internet Imaging II conference*, San José, CA, USA, 2001.

[102] J. L. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, 1995.

[103] C. Y. Han, H. Chen, L. He, and W. G. Wee. A web-based distributed image processing system. In S. Santini and R. Schettini, editors, *Proceedings of the SPIE Photonics West Conference*, volume 5018, pages 111–122, San Jose, CA, USA, 2003. SPIE.

[104] R. Haralick and L. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1993.

[105] R. M. Haralick. Statistical and structural approaches to texture. In *Proceedings of the IEEE Computer Society*, volume 67, pages 786–804, 1979.

[106] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[107] H. L. Harter. Critical values for Duncan's new multiple range test. *Biometrics*, 16(4):671–685, 1960.

[108] K. Hartung, S. Münch, L. Schomaker, T. Guiard-Marigny, B. Le Goff, R. MacLaverty, J. Nijtmans, A. Camurri, I. Defée, and C. Benoît. Di 3 - development of a system architecture for the acquisition, integration, and representation of multimodal information. Technical report, Esprit Project 8579 MIAMI - WP 3, 1996.

[109] K. Hartung, S. Münch, L. Schomaker, T. Guiard-Marigny, B. Le Goff, R. MacLaverty, J. Nijtmans, A. Camurri, I. Defée, and C. Benoît. Di 4 - software architecture. Technical report, Esprit Project 8579 MIAMI - WP 3, 1996.

[110] P. A. van der Helm and E. L. J. Leeuwenberg. Holographic goodness is not that bad: Reply to olivers, chater, and watson (2004). *Psychological Review*, 111(1):261–273, 2004.

[111] H. J. van den Herik and E. O. Postma. *Discovering the visual signature of painters*, volume 45 of *Studies in Fuzziness and Soft Computing*, chapter 7, pages 129–147. Springer Verlag (Physica Verlag), Heidelberg-Tokyo-New York, 2000. ISBN: 3790812765.

[112] F. Hernandez, C. Wert, I. Recio, B. Aguilera, W. Koch, M. Bogensperger, P. Linde, G. Günter, B. Mulrenin, X. Agenjo, R. Yeats, L. Bordoni, and F. Poggi. XML for libraries,

archives, and museums: The projects COVAX. *Applied Artificial Intelligence*, 17(8):797–816, 2003.

[113] E. C. M. Hoenkamp. Unitary operators on the document space. *Journal of the American Society for Information Science and Technology*, 54(4):319–325, 2003.

[114] E. C. M. Hoenkamp and D. Song. The document as an ergodic markov chain. In *Proceedings of the 27th annual international conference on Research and developement in information retrieval*, pages 496–497. ACM Press: New York, NY, USA, 2004.

[115] E. C. M. Hoenkamp, O. Stegeman, and L. R. B. Schomaker. Supporting content retrieval from www via "basic level categories". In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–312. ACM Press: New York, NY, USA, 1999.

[116] E. C. M. Hoenkamp and H. C. van Vugt. The influence of recall feedback in information retrieval on user satisfaction and user behavior. In J. D. Moore and K. Stenning, editors, *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 423–428. University of Edinburgh, Scotland, London: Lawrence Erlbaum Associates, Publishers, August 2001.

[117] L. Hollink, A. Th. Schreiber, B. J. Wielinga, and M. Worring. Classification of user image descriptions. *International Journal of Human-Computer Studies*, 61(5):601–626, 2004.

[118] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In G. Medioni, R. Nevatia, D. Huttenlocher, and J. Ponce, editors, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1997.

[119] J. Huang and R. Zabih. Combining color and spatial information for content-based image retrieval. In *Proceedings of the DARPA Image Understanding Workshop*, pages 687–691, 1997.

[120] W. Humboldt. *Gesammelte werke*. Berlin, 1841–1852.

[121] A. Humrapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. R. Bach, M. Gorkani, and R. C. Jain. Virage video engine. In I. K. Sethi and R. C. Jain, editors, *Storage and Retrieval for Image and Video Databases V*, volume 3022 of *Electronic Imaging: Science and Technology*, pages 188–198, San Jose, CA, USA, 1997.

[122] N. Ikonomakis, K. N. Plataniotis, and A. N. Venetsanopoulos. User interaction in region-based color image segmentation. In D. P. Huijsmans and A. W. M. Smeulders, editors, *Visual Information and Information Systems: Third International Conference (VISUAL'99)*, volume 1614 of *Lecture Notes in Computer Science*, pages 99–106. Springer-Verlag GmbH, 1999.

[123] Google Inc. Google Blog Wednesday, November 10, 2004: Google's index nearly doubles, URL: http://www.google.com/googleblog/2004/11/googles-index-nearly-doubles.html [Last accessed on July 31, 2005].

[124] I. Isgum, B. van Ginneken, and M. Olree. Automatic detection of calcifications in the aorta from CT scans of the abdomen. *Academic Radiology*, 11:247–257, 2004.

[125] M. Israël, E. L. van den Broek, P. van der Putten, and M. J. den Uyl. Automating the construction of scene classifiers for content-based video retrieval. In L. Khan and V. A. Petrushin, editors, *Proceedings of the Fifth ACM International Workshop on Multimedia Data Mining (MDM/KDD'04)*, pages 38–47, Seattle, WA, USA, 2004.

[126] M. Israël, E. L. van den Broek, P. van der Putten, and M. J. den Uyl. Real time automatic scene classification. In R. Verbrugge, N. Taatgen, and L. R. B. Schomaker, editors, *Proceedings of the Sixteenth Belgium-Netherlands Artificial Intelligence Conference*, pages 401–402, 2004.

[127] A. K. Jain and K. Karu. Learning texture discrimination masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):195–205, 1996.

[128] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[129] J. R. Janesick. *Scientific Charge-Coupled Devices*. SPIE - The International Society for Optical Engineering, 2001.

[130] M.-C. Jaulent, C. Le Bozec, Y. Cao, E. Zapletal, and P. Degoulet. A property concept frame representation for flexible image content retrieval in histopathology databases. In *Proceedings of the Annual Symposium of the American Society for Medical Informatics (AMIA)*, Los Angeles, CA, USA, 2000.

[131] C. M. Jonker, J. J. P. Schalken, J. Theeuwes, and J. Treur. Human experiments in trust dynamics. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Trust Management, Proceedings of the Second International Conference on Trust Management*, volume 2995 of *Lecture Notes in Computer Science*, pages 206–220. Springer Verlag, 2004.

[132] C. M. Jonker and J. Treur. An agent architecture for multi-attribute negotiation. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1195–1201, 2001.

[133] C. M. Jonker and J. Treur. Modelling the dynamics of reasoning processes: Reasoning by assumption. *Cognitive Systems Research Journal*, 4:119–136, 2003.

[134] C. M. Jonker, J. Treur, and W. C. A. Wijngaards. An agent-based architecture for multimodal interaction. *International Journal of Human-Computer Studies*, 54(3):351–405, 2001.

[135] C. Jörgensen. Access to pictorial material: A review of current research and future prospects. *Computers and the Humanities*, 33(4):293–318., 1999.

[136] F.-D. Jou, K.-C. Fan, and Y.-L. Chang. Efficient matching of large-size histograms. *Pattern Recognition Letters*, 25(3):277–286, 2004.

[137] T. Kanungo, C. H. Lee, and R. Bradford. What fraction of image on the web contain text? In A. Antonacopoulos and J. Hu, editors, *Proceedings of the First International Workshop on Web Document Analysis (WDA2001)*, Seattle, Washington, USA, 2001. World Scientific Publishing Co.

[138] R. Kasturi, S. H. Strayer, U. Gargi, and S. Antani. An evaluation of color histogram based methods in video indexing. Technical Report CSE-96-053, Department of Computer Science and Engineering, Pennsylvania State University, 1996.

[139] T. Kato. Database architecture for content-based image retrieval. In A. A. Jambardino and W. R. Niblack, editors, *Proceedings of SPIE Image Storage and Retrieval Systems*, volume 1662, pages 112–123, San Jose, CA, USA, February 1992.

[140] P. Kay. The World Color Survey. URL: http://www.icsi.berkeley.edu/wcs/, [Last accessed on July 31, 2004].

[141] P. Kay and W. Kempton. What is the Sapir-Whorf hypothesis? *American Anthropologist*, 86(1):65–79, 1984.

[142] J. Kender and B. Yeo. Video scene segmentation via continuous video coherence. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 367–373, Santa Barbara, CA, june 1998. IEEE Computer Society.

[143] R. Kimmel, N. Kiryati, and A. M. Bruckstein. Multivalued distance maps for motion planning on surfaces with moving obstacles. *IEEE Transactions on Robotics and Automation*, 14(3):427–436, 1998.

[144] R. Kimmel, D. Shaked, N. Kiryati, and A. M. Bruckstein. Skeletonization via distance maps and level sets. *Computer Vision and Image Understanding*, 62(3):382–391, 1995.

[145] I. King and Z. Jin. Integrated probability function and its application to content-based image retrieval by relevance feedback. *Pattern Recognition*, 36(9):2177–2186, 2003.

[146] P. M. F. Kisters. Color based image retrieval: The human centered approach. Master's thesis, Department of Artificial Intelligence, Radboud University Nijmegen, 2005.

[147] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[148] J. J. Koenderink, A. J. van Doorn, A. M. L. Kappers, S. F. te Pas, and S. C. Pont. Il-

lumination direction from texture shading. *Journal of the Optical Society of America A*, 20(6):987–995, 2003.

[149] J. S. Kole and F. J. Beekman. Evaluation of the ordered subset convex algorithm for cone-beam CT. *Physics in Medicine and Biology*, 50:613–623, 2005.

[150] A. Koning and R. J. van Lier. Mental rotation depends on the number of objects rather than on the number of image fragments. *Acta Psychologica*, 117(1):65–77, 2004.

[151] M. Koskela, J. Laaksonen, S. Laakso, and E. Oja. Evaluating the performance of content-based image retrieval systems. In *Proceedings of VISual 2000*, Proceedings of VISual, Lyon, France, November 2000. URL: http://www.cis.hut.fi/pisom/publications.html [Last accessed on July 31, 2005].

[152] S. Y. Kung, M. W. Mak, and S. H. Lin. *Biometric Authentication: A Machine Learning Approach*. Prentice Hall, 2005.

[153] W.-J. Kuo, R.-F. Chang, C. C. Lee, W. K. Moon, and D.-R. Chen. Retrieval technique for the diagnosis od solid breast tumors on sonogram. *Ultrasound in Medicine and Biology*, 28(7), 2002.

[154] J. W. Kwak and N. I. Cho. Relevance feedback in content-based image retrieval system by selective region growing in the feature space. *Signal Processing: Image Communication*, 18(9):787–799, 2003.

[155] W.-C. Lai, C. Chang, E. Chang, K.-T. Cheng, and M. Crandell. Pbir-mm: multimodal image retrieval and annotation. In L. Rowe, B. Merialdo, M. Muhlhauser, K. Ross, and N. Dimitrova, editors, *Proceedings of the tenth ACM international conference on Multimedia*, pages 421–422, 2002.

[156] C. H. C. Leung and H. H. S. Ip. Benchmarking for content-based visual information search. In R. Laurini, editor, *Proceedings of the Fourth International Conference on Visual Information Systems: Advances in Visual Information Systems*, volume 1929 of *Lecture Notes in Computer Science*, pages 442–456, Lyon, France, 2000. Springer-Verlag.

[157] H. Li and A. M. Vossepoel. Generation of the euclidean skeleton from the vector distance map by a bisector decision rule. In D. Goldgof, A. Jain, D. Terzopoulos, and Y.-F. Wang, editors, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 66–71, 1998.

[158] R. J. van Lier, P. A. van der Helm, and E. Leeuwenberg. Integrating global and local aspects of visual occlusion. *Perception*, 23:883–903, 1994.

[159] T. Lin and H.J. Zhang. Automatic video scene extraction by shot grouping. In *Proceedings of the 15th IEEE International Conference on Pattern Recognition*, volume 4, pages 39–42, Barcelona, Spain, 2000.

[160] F. Liu. *Modeling spatial and temporal textures*. PhD thesis, Massachusetts Institute of Technology, 1997.

[161] W. Liu, Z. Su, S. Li, Y. F. Sun, and H. Zhang. A performance evaluation protocol for content-based image retrieval algorithms/systems. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop on Empirical Evaluation in Computer Vision*, Kauai, USA, 2001.

[162] Y. Liu and R. T. Collins. A computational model for repeated pattern perception using frieze and wallpaper groups. In J. Ponce, J. Malik, D. Kriegman, and D. Forsyth, editors, *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, volume 1, pages 1537–1544, Hilton Head Island, South Carolina, June 13–15 2000.

[163] L. R. Long, G. R. Thoma, and L. E. Berman. A prototype client/server application for biomedical text/image retrieval on the Internet. In I. K. Sethi and R. C. Jain, editors, *Proceedings of the SPIE Storage and Retrieval for Image and Video Databases VI*, volume 3312, pages 362–372, San Jose, CA, USA, 1997. SPIE.

[164] J. C. A. van der Lubbe, E. P. van Someren, and M. J. T. Reinders. Dating and authentication of rembrandt's etchings with the help of computational intelligence. In *Proceedings of the International Cultural Heritage Informatics meeting*, pages 485–492, Milan - Italy, 2001.

[165] J. Lucy and R. Shweder. Whorf and his critics: Linguistic and nonlinguistic influences on color memory. *American Anthropologist*, 81:581–615, 1979.

[166] J. A. Lucy. *Grammatical categories and cognition: a case study of the linguistic relativity hypothesis*. Cambridge: Cambridge University Press, 1992.

[167] J. A. Lucy. *The Scope of Linguistic Relativity: An Analysis and Review of Empirical Research*. Cambridge: Cambridge University Press, 1996.

[168] W. Ma and B. Manjunath. Netra: A toolbox for navigating large image databases. In *Proceedings of the IEEE International Conference on Image Processing*, pages 568–571, 1997.

[169] W. Y. Ma and B. S. Manjunath. Netra: a toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[170] P.-P. van Maanen and K. van Dongen. Towards task allocation decision support by means of cognitive modeling of trust. In C. Castelfranchi, S. Barber, J. Sabater, and M. Singh, editors, *Proceedings of the AAMAS-05 workshop Trust in Agent Societies (Trust)*, page [in press], Utrecht - The Netherlands, 2005.

[171] S. D. MacArthur, C. E. Brodley, A. C. Kak, and L. S. Broderick. Interactive content-

based image retrieval using relevance feedback. *Computer Vision and Image Understanding*, 88(2):55–75, 2002.

[172] T. Mäenpää and M. Pietikäinen. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629–1640, 2004.

[173] A. Maerz and M. R. Paul. *A dicitonary of color*. New York: McGraw-Hill.

[174] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer Verlag, 2003.

[175] M. K. Mandal, S. Panchanathan, and T. Aboulnasr. Illumination invariant image indexing using moments and wavelets. *Journal of Electronic Imaging*, 7(2):282–293, 1998.

[176] S. Marchand-Maillet. Performance evaluation in cbir: the benchathlon network. In N. Boujemaa, C.-S. Li, and D. Forsyth, editors, *Proceedings of MultiMedia Content Based Indexing and Retrieval Workshop 2001*, pages 107–110, Paris, France, September 2001.

[177] Stéphane Marchand-Maillet. The Benchathlon Network: Home of CBIR benchmarking. URL: http://www.benchathlon.net/, [Last accessed on July 31, 2005].

[178] Massachusetts Institute of Technology. Vision Texture. URL: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html, [Last accessed on July 31, 2005].

[179] M. E. Mattie, L. Staib, E. Stratmann, H. D. Tagare, J. Duncan, and P. L. Miller. PathMaster: Content-based cell image retrieval using automated feature extraction. *Journal of the American Medical Informatics Association*, 7(4):404–415, 2000.

[180] J.C. Meadows. Disturbed perception of colours associated with localized cerebral lesions. *Brain*, 97:615–632, 1974.

[181] Merriam-Webster, Incorporated. Merriam-Webster Online. URL: http://www.m-w.com/, [Last accessed on July 31, 2005].

[182] F. Meyer. Automatic screening of cytological specimens. *Computer Vision, Graphics and Image Processing*, 35:356–369, 1986.

[183] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38:113–125, 1994.

[184] T. P. Minka. An image database browser that learns from user interaction. Technical Report 365, MIT Media Laboratory, 1996.

[185] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler. A review of content-based image retrieval systems in medicine - clinical benefits and future directions. *International Journal of Medical Informatics*, 73:1–23, 2004.

[186] H. Müller, W. Müller, S. Marchand-Mallet, T. Pun, and D. McG. Squire. A framework for benchmarking in cbir. *Multimedia Tools and Applications*, 21(1):55–73, 2003.

[187] H. Müller, W. Müller, D. McG. Squire, Stéphane Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters*, 22(5):593–601, 2001.

[188] H. Müller, A. Rosset, J.-P. Vallée, and A. Geissbuhler. Comparing feature sets for content-based medical information retrieval. In *Proceedings of SPIE Medical Imaging*, San Diego, CA, USA, 2004.

[189] B. H. Murray. Sizing the internet. Technical report, Cyveillance, Inc., 2000.

[190] T. Myer. Card sorting and cluster analysis. Technical report, IBM developerWorks, 2001.

[191] NetCraft Ltd. Web server survey statistics. URL: http://news.netcraft.com/archives/web_server_survey.html, [Last accessed on July 31, 2005].

[192] S. Newsam, L. Wang, S. Bhagavathy, and B. S. Manjunath. Using texture to analyze and manage large collections of remote sensed image and video data. *Journal of Applied Optics: Information Processing*, 43(2):210–217, 2004.

[193] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutos. The QBIC project: Querying images by content using color, texture, and shape. In W. Niblack, editor, *Proceedings of Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, February 1993.

[194] W. Niblack, X. Zhu, J. L. Hafner, T. Breuel, D. B. Ponceleón, D. Petkovic, M. Flickner, E. Upfal, S. I. Nin, S. Sull, B. Dom, B.-L. Yeo, S. Srinivasan, D. Zivkovic, and M. Penner. Updates of the qbic system. In I. K. Sethi and R. Jain, editors, *Proceedings of SPIE Storage and Retrieval for Image and Video Databases VI*, volume 3312, pages 150–161, San Jose, CA, USA, January 1998. SPIE.

[195] R. Niels and L. G. Vuurpijl. Using dynamic time warping for intuitive handwriting recognition. In A. Marcelli and C. De Stefano, editors, *Proceedings of the 12th Conference of the International Graphonomics Society (IGS2005)*, page [in press], Salerno, Italy, 2005.

[196] J. Nijtmans. Multimodal integration for advanced multimedia interfaces, URL: http://hwr.nici.kun.nl/~miami/ [Last accessed on July 31, 2005].

[197] J. M. Nyce and P. Kahn. *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*. Academic Press, Inc. (Hartcourt Brace Jovanovich Publishers), 1991.

[198] M. R. Ogiela and R. Tadeusiewicz. Semantic-oriented syntactic algorithms for content recognition and understanding of images in medical databases. In *Proceedings of the*

Second International Conference on Multimedia and Exposition (ICME2001), pages 621–624, Tokyo, Japan, 2001. IEEE Computer Society.

[199] P. P. Ohanian and R. C. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8):819–833, 1992.

[200] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 1, pages 701–706, Quebec, Canada, 2002.

[201] T. Ojala and M. Pietikäinen. Unsupervised texture segmentation using feature distribution. *Pattern Recognition*, 32(3):477–486, 1999.

[202] Optical Society of America, Committee on Colorimetry. *The science of color*. New York: Crowell, 1953.

[203] E. den Os. COMIC - COnversational Multimodal Interaction with Computers, URL: http://www.hcrc.ed.ac.uk/comic/ [Last accessed on July 31, 2005].

[204] C. Osgood. *An exploration into semantic space*. Human Communication. New York: Basic Books, 1963.

[205] C. Palm. Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5):965–976, 2004.

[206] B. Pascal. *Pascal's pensées*. London: Routledge & Kegan Paul, 1950.

[207] J. S. Payne, L. Hepplewhite, and T. J. Stoneham. Applying perceptually-based metrics to textural image retrieval methods. In B. E. Rogowitz and T. N. Pappas, editors, *Proceedings of Human Vision and Electronic Imaging V*, volume 3959, pages 423–433, San Jose, CA, USA, 2000.

[208] J. Penn. *Linguistic relativity versus innate ideas: The origins of the Sapir-Whorf hypothesis in German thought*. Paris: Mounton, 1972.

[209] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content based manipulation of image databases. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases II*, volume 2185 of *Electronic Imaging: Science and Technology*, pages 34–47, San Jose, CA, USA, 1994.

[210] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[211] M. Petković and W. Jonker. *Content-Based Video Retrieval: A Database Perspective*, volume 25 of *Multimedia systems and applications*. Kluwer Academic Publishers, Boston, 2003.

[212] T. V. Pham, M. Worring, and A. W. M. Smeulders. Face detection by aggregated bayesian networks. *Pattern Recognition Letters*, 23(4):451–461, 2002.

[213] I. T. Phillips and A. K. Chhabra. Empirical performance evaluation of graphics recognition systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):849–870, 1999.

[214] I. T. Phillips, J. Liang, A. Chhabra, and R. M. Haralick. *A Performance Evaluation Protocol for CBIR Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 372–389. 1998.

[215] S. C. Pont and J. J. Koenderink. Bidirectional texture contrast function. *International Journal of Computer Vision*, 62(1-2):17–34, 2005.

[216] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *Proceedings the IEEE International Conference on Computer Vision*, volume 2, pages 1165–1173, Corfu, Greece, september 1999.

[217] K. Ravishankar, B. Prasad, S. Gupta, and K. Biswas. Dominant color region based indexing for cbir. In V. Roberto, V. Cantoni, and S. Levialdi, editors, *Proceedings of the International Conference on Image Analysis and Processing*, pages 887–892. Italian Chapter of the International Association for Pattern Recognition (IAPR-IC), september 1999.

[218] A. Ravishankar Rao and G. L. Lohse. Towards a texture naming system: Identifying relevant dimensions of texture. *Vision Research*, 36(11):1649–1669, 1996.

[219] T. Regier, P. Kay, and R. S. Cook. Focal colors are universal after all. *Proceedings of the National Academy of Sciences USA*, 102(23):8386–8391, 2005.

[220] R. A. Rensink. Grouping in visual short-term memory [abstract]. *Journal of Vision*, 1(3):126a, 2001.

[221] Rijksmuseum. Research, URL: http://www.rijksmuseum.nl/wetenschap/ [Last accessed on July 31, 2005].

[222] Rijksmuseum. Search for professionals, URL: http://www.rijksmuseum.nl/wetenschap/zoeken [Last accessed on July 31, 2005].

[223] E. M. van Rikxoort and E. L. van den Broek. Texture analysis. Technical report, NICI, Radboud University Nijmegen; URL: http://www.few.vu.nl/~egon/publications/pdf/Rikxoort04-Texture-analysis.pdf [Last accessed on July 31, 2005], 2004.

[224] E. M. van Rikxoort, E. L. van den Broek, and Th. E. Schouten. The development of a human-centered object-based image retrieval engine. In B. J. A. Kröse, H. J. Bos, E. A. Hendriks, and J. W. J. Heijnsdijk, editors, *Proceedings of the Eleventh Annual Conference*

of the Advanced School for Computing and Imaging, pages 401–408, The Netherlands - Heijen, 2005.

[225] E. M. van Rikxoort, E. L. van den Broek, and Th. E. Schouten. Mimicking human texture classification. *Proceedings of SPIE (Human Vision and Electronic Imaging X)*, 5666:215–226, 2005.

[226] D. Roberson. Color categories are culturally diverse in cognition as well as in language. *Cross-Cultural Research: The Journal of Comparative Social Science*, 39:56–71, 2005.

[227] D. Roberson, J. Davidoff, I. R. L. Davies, and L. R. Shapiro. Color categories: Evidence for the cultural relativity hypothesis. *Cognitive Psychology*, 50(4):378–411, 2005.

[228] D. Roberson and C. O'Hanlon. How culture might constrain colour categories. *Behavioral and Brain Sciences*, -, (in press).

[229] G. P. Robinson, H. D. Tagare, J. S. Duncan, and C. C. Jaffe. Medical image collection indexing: Shape-based retrieval using KD-trees. *Computerized Medical Imaging and Graphics*, 20(4):209–217, 1996.

[230] Rochester Institute of Technology. Munsell Color Science Laboratory. URL: `http://www.cis.rit.edu/mcsl/`, [Last accessed on July 31, 2005].

[231] K. Rodden. How do people organise their photographs? In *Proceedings of the 21st Annual Colloquium on Information Retrieval Research*, Electronic Workshops in Computing, (eWiC), Glasgow, Scotland, April 19-20 1999. British Computer Society.

[232] E. Rosch Heider. Universals in color naming and memory. *Journal of Experimental Psychology*, 93(1):10–20, 1972.

[233] A. Rosenfeld and J. L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[234] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 699–704, Madison, Wisconsin, 2003.

[235] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Past, present, and future. *Journal of Visual Communication and Image Representation*, 10:1–23, 1999.

[236] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Information retrieval beyond the text document. *Library Trend*, 48(2):437–456, 1999.

[237] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998.

[238] J. P. de Ruiter, S. Rossignol, L. G. Vuurpijl, D. W. Cunningham, and W. J. M. Levelt. Slot: A research platform for investigating multimodal communication. *Behavior Research Methods, Instruments, & Computers*, 35(3):408–419, 2003.

[239] O Sacks. *An anthropologist on Mars*. New York: Knopf, 1995.

[240] E. Sapir. The status of linguistics as a science. *Language*, 5:209, 1929.

[241] A. Sbober, C. Eccher, E. Blanzieri, P. Bauer, M. Cristifolini, G. Zumiani, and S. Forti. A multiple classifier system for early melanoma diagnosis. *Artificial Intelligence in Medicine*, 27:29–44, 2003.

[242] R. Schettini, G. Ciocca, and S. Zuffi. *A survey of methods for colour image indexing and retrieval in image databases*. J. Wiley, 2001.

[243] B. Schiele. How many classifiers do I need? In R. Kasturi, D. Laurendeau, and C. Suen, editors, *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 176–179. IEEE Computer Society, 2002.

[244] I. M. Schlesinger. *The wax and wane of Whorfian views*. New York: Mounton de Gruyter, 1991.

[245] P. Schmidt-Saugeon, J. Guillod, and J.-P. Thiran. Towards a computer-aided diagnosis system for pigmented skin lesions. *Computerized Medical Imaging and Graphics*, 27:65–78, 2003.

[246] L. Schomaker, E. de Leau, and L. Vuurpijl. Using pen-based outlines for object-based annotation and image-based queries. In D. P. Huijsmans and A. W. M. Smeulders, editors, *Third International Conference on Visual Information and Information Systems (VISUAL)*, volume 1614 of *Lecture Notes in Computer Science*, pages 585–592, Amsterdam, The Netherlands, June 2–4 1999.

[247] L. Schomaker, J. Nijtmans, A. Camurri, F. Lavagetto, P. Morasso, C. Benoît, T. Guiard-Marigny, B. Le Goff, J. Robert-Ribes, A. Adjoudani, I. Defée, S. Münch, K. Hartung, and J. Blauert. Di 2 - progress report. Technical report, Esprit Project 8579 MIAMI - WP 2, 1995.

[248] L. Schomaker, J. Nijtmans, A. Camurri, F. Lavagetto, P. Morasso, C. Benoît, T. Guiard-Marigny, B. Le Goff, J. Robert-Ribes, A. Adjoudani, I. Defée, S. Münch, K. Hartung, and J. Blauert. A taxonomy of multimodal interaction in the human information processing system. Technical report, Esprit Project 8579 MIAMI - WP 1, 1995.

[249] L. Schomaker, L. Vuurpijl, and E. de Leau. New use for the pen: outline-based image queries. In *Proceedings of the 5th IEEE International Conference on Document Analysis*, pages 293–296, Piscataway (NJ), USA, 1999.

[250] L. R. B. Schomaker and L. G. Vuurpijl. *IWFHR VII - Seventh International Workshop on Frontiers in Handwriting Recognition Proceedings*. Amsterdam, The Netherlands, September 2000.

[251] Th. E. Schouten, H. C. Kuppens, and E. L. van den Broek. Three dimensional fast exact euclidean distance (3D-FEED) maps. In *[submitted]*, 2006.

[252] Th. E. Schouten, H. C. Kuppens, and E. L. van den Broek. Video surveillance using distance maps. In *[submitted]*, 2006.

[253] Th. E. Schouten, H. C. Kuppens, and E. L. van den Broek. Timed Fast Exact Euclidean Distance (tFEED) maps. *Proceedings of SPIE (Real Time Electronic Imaging X)*, 5671:52–63, 2005.

[254] Th. E. Schouten and E. L. van den Broek. Fast Exact Euclidean Distance (FEED) Transformation. In J. Kittler, M. Petrou, and M. Nixon, editors, *Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR 2004)*, volume 3, pages 594–597, Cambridge, United Kingdom, 2004.

[255] S. Sclaroff, L. Taycher, and M. la Cascia. Imagerover: A content-based image browser for the world wide web. In R. W. Picard, F. Liu, G. Healey, M. Swain, and R. Zabih, editors, *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 2–9, 1997.

[256] J. H. Seppenwoolde, J. F. W. Nijsen, L. W. Bartels, S. W. Zielhuis, A. D. van het Schip, and C. J. G. Bakker. Qualitative and quantitative mr imaging of the biodistribution of holmium-loaded microspheres in animal models. *Magnetic Resonance in Medicine*, 53:76–84, 2005.

[257] M. Sharma and S. Singh. Evaluation of texture methods for image analysis. In R. Linggard, editor, *Proceedings of the 7th Australian and New Zealand Intelligent Information Systems Conference*, pages 117–121, Perth, Western Australia, 2001. ARCME.

[258] F. Y. Shih and J. J. Liu. Size-invariant four-scan euclidean distance transformation. *Pattern Recognition*, 31(11):1761–1766, 1998.

[259] F. Y. Shih and Y.-T. Wu. Fast euclidean distance transformation in two scans using a 3×3 neighborhood. *Computer Vision and Image Understanding*, 93(2):195–205, 2004.

[260] M.-Y. Shih and D.-C. Tseng. A wavelet-based multiresolution edge detection and tracking. *Image and Vision Computing*, 23(4):441–451, 2005.

[261] Y. Shirai. Reply performance characterization in computer vision. *Computer Vision, Graphics, and Image Processing - Image Understanding*, 60(2):260–261, 1994.

[262] B. Shneiderman and H. Kang. Direct annotation: A drag-and-drop strategy for label-

ing photos. In *Proceedings of the IEEE International Conference on Information Visualization*, pages 88–98, London, England, July 19 - 21 2000.

[263] C.-R. Shyu, C. E. Brodley, A. C. Kak, A. Kosaka, A. M. Aisen, and L. S. Broderick. ASSERT: A physician-in-the-loop content-based retrieval system for HRCT image databases. *Computer Vision and Image Understanding*, 75(1-2):111–132, 1999.

[264] J. Simons. Using a semantic user model to filter the World Wide Web proactively. In A. Jameson, C. Paris, and C. Tasso, editors, *Proceedings of the Sixth International Conference, User Modeling*, pages 455–456, Sardinia, Italy, 1997. Springer: Wien, New York.

[265] J. Simons. Profile - a Proactive Information Filter, URL: http://hwr.nici.kun.nl/~profile/ [Last accessed on July 31, 2005].

[266] J. Simons, A. T. Arampatzis, B. C. M. Wondergem, L. R. B. Schomaker, P. van Bommel, E. C. M. Hoenkamp, Th. P. van der Weide, and C. H. A. Koster. PROFILE - a multidisciplinary approach to information discovery. Technical Report CSI-R0001, Institute for Computing and Information Science, Radboud University Nijmegen, 2000.

[267] M. Singh, M. Markou, and S. Singh. Colour image texture analysis: Dependence on colour space. In C. Y. Suen, R. Kasturi, and R. Plamondon, editors, *Proceedings of the 16th IEEE International Conference on Pattern Recognition*, volume 1, pages 672–676, Quebec City, QC, Canada, 2002.

[268] D. Slobin. *From "thought and language" to "thinking and speaking"*. Cambridge: Cambridge University Press, 1996.

[269] P. A. M. Smeets, C. de Graaf, A. Stafleu, M. J. P. van Osch, and J. van der Grond. Functional MRI of human hypothalamic responses following glucose ingestion. *NeuroImage*, 24:363–368, 2005.

[270] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[271] J. G. Smirniotopoulos. MedPix^{TM} Medical Image Database, URL: http://rad.usuhs.mil/medpix/ [Last accessed on July 31, 2005].

[272] G. Smith. Meastex image texture database and test suite. URL: http://www.cssip.uq.edu.au/meastex/meastex.html, [Last accessed on July 31, 2005].

[273] J. R. Smith. *Integrated spatial and feature image systems: Retrieval, analysis, and compression*. PhD thesis, Columbia University, 1997.

[274] J. R. Smith and S.-F. Chang. Single color extraction and image query. In B. Liu, editor,

Proceedings of the 2nd IEEE International Conference on Image Processing, pages 528–531. IEEE Signal Processing Society, IEEE Press, 1995.

[275] J. R. Smith and S. F. Chang. Tools and techniques for color image retrieval. In I. K. Sethi andR. C. Jain, editor, *Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV*, volume 2760 of *Electronic Imaging*. San Jose, CA: IS&T/SPIE, 1996.

[276] J. R. Smith and S. F. Chang. *Querying by color regions using the VisualSEEk content-based visual query system*, chapter 2, pages 23–42. The AAAI Press, 1997.

[277] C. G. M. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35, 2005.

[278] P. R. Snoeren and N. Karssemeijer. Thickness correction of mammographic images by means of a global parameter model of the compressed breast. *IEEE Transactions on Medical Imaging*, 23(7):799–806, 2004.

[279] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision*. PWS publishing, San Francisco, USA, 1999.

[280] D. McG. Squire, W. Müller, and H. Müller. Relevance feedback and term weighting schemes for content-based image retrieval. In D. P. Huismans and A. W. M. Smeulders, editors, *Proceedings of the Third International Conference on Visual Information Systems*, pages 549–556, Amsterdam, The Netherlands, 1999. Springer-Verlag.

[281] J. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23:501–509, 2004.

[282] M. van Staalduinen, J. C. A. van der Lubbe, and E. Backer. Circular analysis based line detection filters for watermark extraction in x-ray images of etchings. In J. van Wijk, R. Veltkamp, and K. Langendoen, editors, *Proceedings of the Tenth Annual Conference of the Advanced School for Computing and Imaging*, pages 305–310, Ouddorp - The Netherlands, 2004.

[283] N. Steiner. A review of web based color pickers. URL: http://www.web-graphics.com/feature-002.php [Last accessed on July 31, 2005], 2002.

[284] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet - srgb. Technical report, W3C; http://www.w3.org/Graphics/Color/sRGB.html [Last accessed on July 31, 2005], 1996.

[285] M. A. Stricker and M. J. Swain. The capacity of color histogram indexing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 704–708, Madison, Wisconsin, 1994.

[286] J. Sturges and T. W. A. Whitfield. Locating basic colours in the munsell space. *Color Research and Application*, 20:364–376, 1995.

[287] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[288] Genootschap Onze Taal. *Onze Taal Taalkalender*. Den Haag: SDU, 2003.

[289] J. H. Takala and J. O. Viitanen. Distance transform algorithm for Bit-Serial SIMD architectures. *Computer Vision and Image Understanding*, 74(2):150–161, 1999.

[290] H. Y. Tang, Lilian, R. Hanka, H. H. S. Ip, K. K. T. Cheung, and R. Lam. Semantic query processing and annotation generation for content-based retrieval of histological images. In *International Symposium on Medical Imaging*, volume 3976, San Diego, CA, USA, 2000. SPIE.

[291] M. J. Tarr, H. H. Bülthoff, M. Zabinski, and V. Blanz. To what extent do unique parts influence recognition across changes in viewpoint? *Psychological Science*, 8(4):282–289, 1997.

[292] The MathWorks, Inc. The MATLAB Central File Exchange, URL: http://www.mathworks.com/matlabcentral/fileexchange/ [Last accessed on July 31, 2005].

[293] The Rijksmuseum, the Leiden Print Room and other Dutch public collections. Early Photography 1839–1860, URL: http://www.earlyphotography.nl [Last accessed on July 31, 2005].

[294] The W3C Validator Team. The W3C Markup Validation Service v0.6.7, URL: http://validator.w3.org [Last accessed on July 31, 2005].

[295] E. L. Thorndike and I. Lorge. *The techer's word book of 30,000 words*. New York: Techers College, Columbia University, 1944.

[296] J. Trant. *Image Retrieval Benchmark Database Service: A Needs Assessment and Preliminary Development Plan*. Archives & Museum Informatics, Canada, 2004.

[297] S. Treue. Perceptual enhancement of contrast by attention. *Trends in Cognitive Sciences*, 8(10):435–437, 2004.

[298] R. C. Tryon. *Cluster Analysis*. Ann Arbor, Michigan: Edwards Brothers Inc., 1939.

[299] M. Tuceryan and A. K. Jain. *Texture Analysis*, chapter 2.1, pages 235–276. World Scientific, 1993.

[300] K. Valkealahti and E. Oja. Reduced multidimensional histograms in color texture description. In *Proceedings of the 14th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 1057–1061, Brisbane, Australia, 1998.

[301] E. N. van Lin, L. P. van der Vight, J. A. Witjes, H. J. Huisman, J. W. Leer, and A. G. Visser. The effect of an endorectal balloon and off-line correction on the interfraction systematic and random prostate position variations: A comparative study. *International Journal of Radiation Oncology Biology Physics*, 61(1):278–288, 2004.

[302] R. Veltkamp, H. Burkhardt, and H. Kriegel. *State-of-the-Art in Content-Based Image and Video Retrieval*. Kluwer Academic Publishers, 2001.

[303] R. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical report, Department of Computing Science, Utrecht University, 2000. http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/ [Last accessed on July 31, 2005].

[304] J. Vendrig and M. Worring. Interactive adaptive movie annotation. *IEEE Multimedia*, (july/september):30–37, 2003.

[305] C.C. Venters and M. Cooper. A review of content-based image retrieval systems. internal report, JTAP, 2000. http://www.jtap.ac.uk/reports/htm/jtap-054.html [Last accessed on July 31, 2005].

[306] H. von Helmholtz. *Die lerne van den Tonempfindungen als physiologische grundlage fur die theorie der musik*. Braunschweig: Friedrich Vieweg und sohn, 5 edition, 1896. Original work published in 1864.

[307] P. de Vrieze, P. van Bommel, Th. P. van der Weide, and J. Klok. Adaptation in multimedia systems. *Multimedia Tools and Applications*, 25(3):333–343, 2005.

[308] L. Vuurpijl, L. Schomaker, and E. L. van den Broek. Vind(x): Using the user through cooperative annotation. In S. N. Srihari and M. Cheriet, editors, *Proceedings of the Eighth IEEE International Workshop on Frontiers in Handwriting Recognition*, pages 221–226, Ontario, Canada, 2002. IEEE Computer Society, Los Alamitos, CA.

[309] W3C. W3C Link Checker v4.1, URL: http://validator.w3.org/checklink/ [Last accessed on July 31, 2005].

[310] J. Wagemans. Detection of visual symmetries. *Spatial Vision*, 9(1):9–32, 1995.

[311] Z. Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[312] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[313] P. Werkhoven. *Multimedia door een sleutelgat*. Amsterdam: Vossiuspers UvA, 2003.

[314] P. Werkhoven, G. Sperling, and C. Chubb. The dimensionality of texture-defined motion: A single channel theory. *Vision Research*, 33(4):463–485, 1993.

[315] P. J. Werkhoven, J. M. Schraagen, and P. A. J. Punte. Seeing is believing: Communication performance under isotropic teleconferencing conditions. *Displays*, 22(4):137–149, 2001.

[316] B. Whorf. *Science and linguistics*. Cambridge, MA: MIT Press, 1940.

[317] B. Whorf. *Language, thought, & reality: Selected writings of Benjamin Lee Whorf*. Cambridge, MA: MIT Press, 1956. reprint of "Science and Language".

[318] B. L. Whorf. *Language, thought, and reality: Selected papers of Benjamin Lee Whorf*, pages 233–245. Cambridge, M.A.: MIT Press, 1956. [Original work published 1941].

[319] P. Wilken and W. J. Ma. A detection theory account of visual short-term memory for color. *Journal of Vision*, 4(8):150a, 2004.

[320] O. Wink, W. J. Niessen, and M. A. Viergever. Multiscale vessel tracking. *IEEE Transactions on Medical Imaging*, 23:130–133, 2004.

[321] M. Worring and Th. Gevers. Interactive retrieval of color images. *International Journal of Image and Graphics*, 1(3):387–414, 2001.

[322] C.-H. Yao and S.-Y. Chen. Retrieval of translated, rotated and scaled color textures. *Pattern Recognition*, 36(4):913–929, 2003.

[323] M. van Zaanen and G. de Croon. Multi-modal information retrieval using FINT. In *Proceedings of the ImageCLEF Workshop*, page [in press], Bath - U.K., 2004.

[324] A. Zelinsky. A mobile robot navigation exploration algorithm. *IEEE Transactions of Robotics and Automation*, 8(6):707–717, 1992.

[325] D. S. Zhang and G. Lu. Evaluation of similarity measurement for image retrieval. In *Proceedings of IEEE International Conference on Neural Networks & Signal Processing*, pages 928–931, Nanjing, China, 2003.

[326] H. Zhang, Z. Chen, M. Li, and Z. Su. Relevance feedback and learning in content-based image search. *World Wide Web*, 6(2):131–155, 2003.

[327] R. Zhang and Z. M. Zhang. A robust color object analysis approach to efficient image retrieval. *EURASIP Journal on Applied Signal Processing*, 4(6):871–885, 2004.

A

Color LookUp Table (CLUT) markers

Table A.1: R,G,B-markpoints for a Color LookUp Table (CLUT) independent of the underlying cognitive task and the points that were different categorized in the color discrimination experiment and in the color memory experiment. Note that this table is spread over two pages.

Color	According to Discrimination and Memory experiment					Discrimination	Memory
blue	000,000,051	000,000,102	000,000,153	000,000,204	000,000,255	051,204,153	000,051,051
	000,051,102	000,051,153	000,051,204	000,051,255	000,102,102	153,000,255	051,000,051
	000,102,153	000,102,204	000,102,255	000,153,153	000,153,204	153,051,255	
	000,153,255	000,204,204	000,204,255	000,255,204	000,255,255		
	051,000,102	051,000,153	051,000,204	051,000,255	051,051,102		
	051,051,153	051,051,204	051,051,255	051,102,102	051,102,153		
	051,102,204	051,102,255	051,153,153	051,153,204	051,153,255		
	051,204,204	051,204,255	051,255,204	051,255,255	102,000,204		
	102,000,255	102,051,204	102,051,255	102,102,153	102,102,204		
	102,102,255	102,153,153	102,153,204	102,153,255	102,204,204		
	102,204,255	102,255,204	102,255,255	153,102,255	153,153,204		
	153,153,255	153,204,204	153,204,255	153,255,204	153,255,255		
	204,204,255	204,255,255					
brown	051,000,000	102,000,000	102,051,000	102,051,051	153,000,000	102,000,051	102,102,051
	153,051,000	153,051,051	153,102,000	153,102,051	153,102,102	153,000,051	153,153,051
	204,000,051	204,051,000	204,102,000	204,102,051	204,102,102	153,000,051	153,153,051
	204,153,102					153,000,051	153,153,102
						153,000,051	204,153,000
						153,051,102	204,153,051
						204,051,051	255,204,153
						204,051,102	102,102,153
						255,102,051	
						255,102,102	
						255,153,102	
yellow	153,153,000	153,153,051	204,204,000	204,204,051	204,204,102	153,153,102	204,153,000
	204,204,051	204,204,102	204,204,153	204,255,000	204,255,051	153,204,000	204,153,051
	204,255,102	204,255,153	255,204,000	255,204,051	255,204,102	153,255,000	255,204,153
	255,255,000	255,255,051	255,255,102	255,255,153	255,255,204	153,204,000	
						153,255,000	
						153,255,051	
gray	000,051,051	051,051,051	102,102,102	102,153,153	153,153,102		102,102,153
	153,153,153	153,153,204	153,204,204	204,204,153	204,204,204		
green	000,051,000	000,051,051	000,102,000	000,102,051	000,102,102		204,204,000
	000,153,000	000,153,051	000,153,102	000,153,153	000,204,000		204,204,051
	000,204,051	000,204,102	000,204,153	000,255,000	000,255,051		
	000,255,102	000,255,153	000,255,204	051,051,000	051,102,000		
	051,102,051	051,102,102	051,153,000	051,153,051	051,153,102		
	051,153,153	051,204,000	051,204,051	051,204,102	051,204,153		
	051,255,000	051,255,051	051,255,102	051,255,153	051,255,204		
	102,102,000	102,102,051	102,153,000	102,153,051	102,153,102		
	102,153,153	102,204,000	102,204,051	102,204,102	102,204,153		
	102,255,000	102,255,051	102,255,102	102,255,153	102,255,204		
	153,153,000	153,153,051	153,153,102	153,204,000	153,204,051		
	153,204,102	153,204,153	153,255,000	153,255,051	153,255,102		
	153,255,153	153,255,204	204,204,102	204,204,153	204,255,000		
	204,255,051	204,255,102	204,255,153	204,255,204			

Color	According to Discrimination and Memory experiment					Discrimination	Memory
orange	153,102,000	204,051,000	204,102,000	204,102,051	204,153,000	153,102,051	153,051,000
	204,153,051	204,153,102	255,102,000	255,102,051	255,153,000	153,153,051	255,051,000
	255,153,051	255,153,102	255,204,000	255,204,051	255,204,102	204,204,000	
	255,204,153					204,204,102	
purple	000,000,051	051,000,051	051,000,102	102,000,051	102,000,102	000,000,102	051,000,153
	102,000,153	102,000,204	102,051,102	102,051,153	102,051,204	051,051,102	051,051,102
	102,102,153	153,000,051	153,000,102	153,000,153	153,000,204	255,102,153	102,051,051
	153,000,255	153,051,102	153,051,153	153,051,204	153,051,255		102,051,255
	153,102,153	153,102,204	153,102,255	153,153,204	153,153,255		153,102,102
	204,000,102	204,000,153	204,000,204	204,000,255	204,051,102		255,051,102
	204,051,153	204,051,204	204,051,255	204,102,153	204,102,204		255,204,255
	204,102,255	204,153,204	204,153,255	204,204,255	255,000,102		
	255,000,153	255,000,204	255,000,255	255,051,153	255,051,204		
	255,051,255	255,102,204	255,102,255	255,153,255			
red	153,000,000	153,000,051	204,000,000	204,000,051	204,051,000	204,000,102	102,000,000
	204,051,051	204,051,102	255,000,000	255,000,051	255,000,102	255,000,153	102,000,051
	255,051,000	255,051,051	255,051,102	255,102,102		255,102,051	153,051,051
							153,102,102
							204,102,102
pink	153,102,102	204,000,102	204,000,153	204,051,102	204,051,153	204,153,255	153,000,102
	204,051,204	204,102,102	204,102,153	204,102,204	204,153,153		153,051,102
	204,153,204	255,000,102	255,000,153	255,000,204	255,000,255		204,000,051
	255,051,102	255,051,153	255,051,204	255,051,255	255,102,102		204,000,204
	255,102,153	255,102,204	255,102,255	255,153,102	255,153,153		204,051,051
	255,153,204	255,153,255	255,204,153	255,204,204	255,204,255		255,000,051
							255,051,051
white	204,204,153	204,204,204	204,255,204	204,255,255	255,255,204	153,204,204	
	255,255,255					153,255,204	
						153,255,255	
						204,204,255	
black	000,000,000	000,000,051	051,000,000	051,051,051		000,051,051	
						051,051,000	

B

Color figures

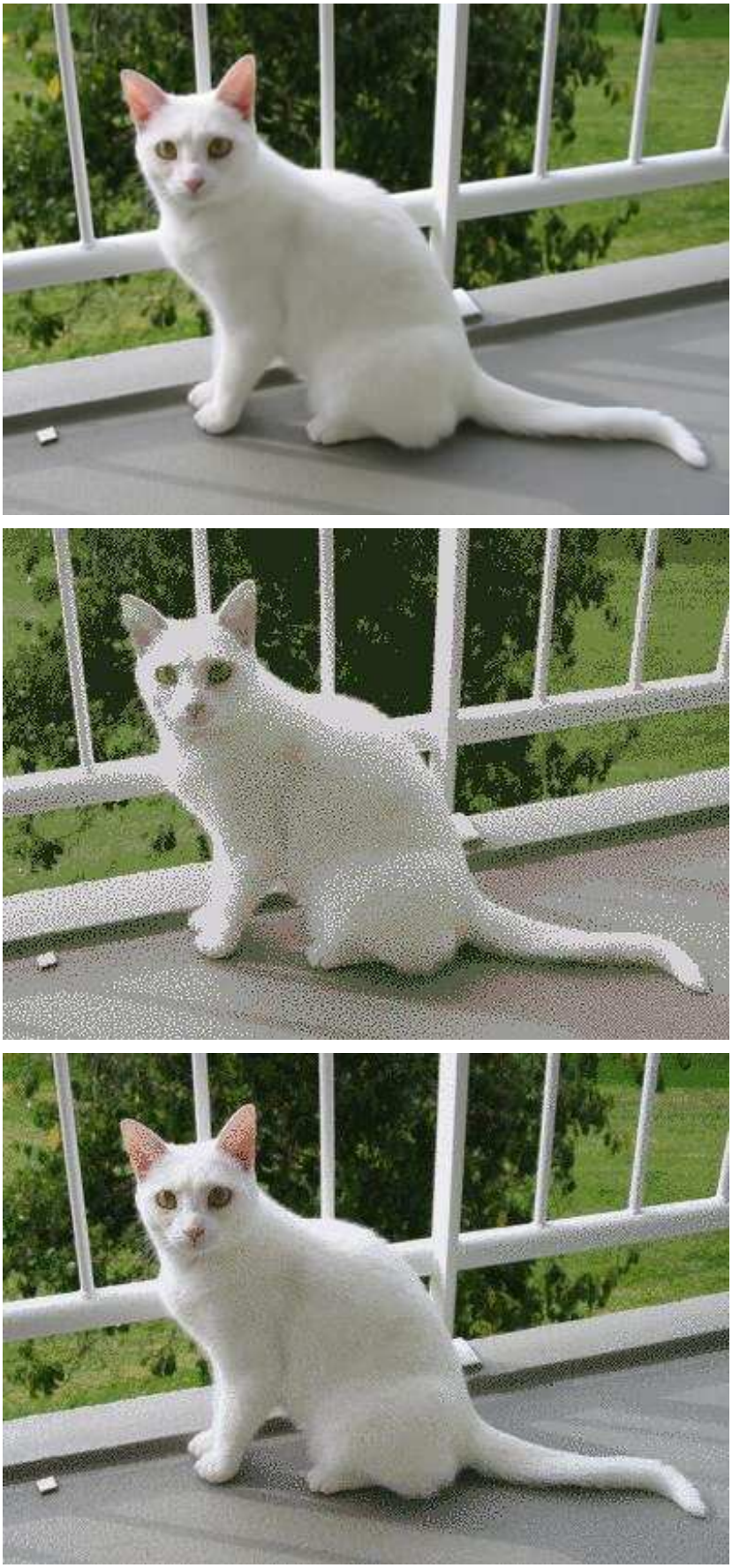


Figure B.1: From top to bottom: The original photo using 256^3 colors, quantized in 8 bins, and quantized in 64 bins, using RGB color space.

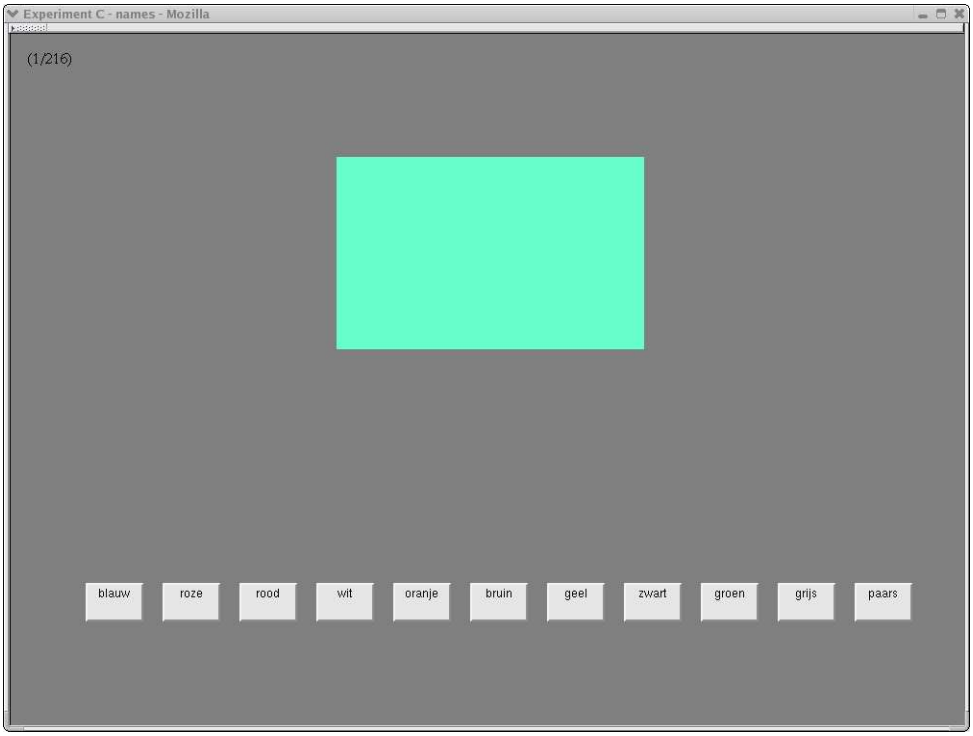


Figure B.2: Screendump of the user interface of the color memory experiment. The buttons were gray and labeled with a color name. So, color classification had to be employed based on color memory.

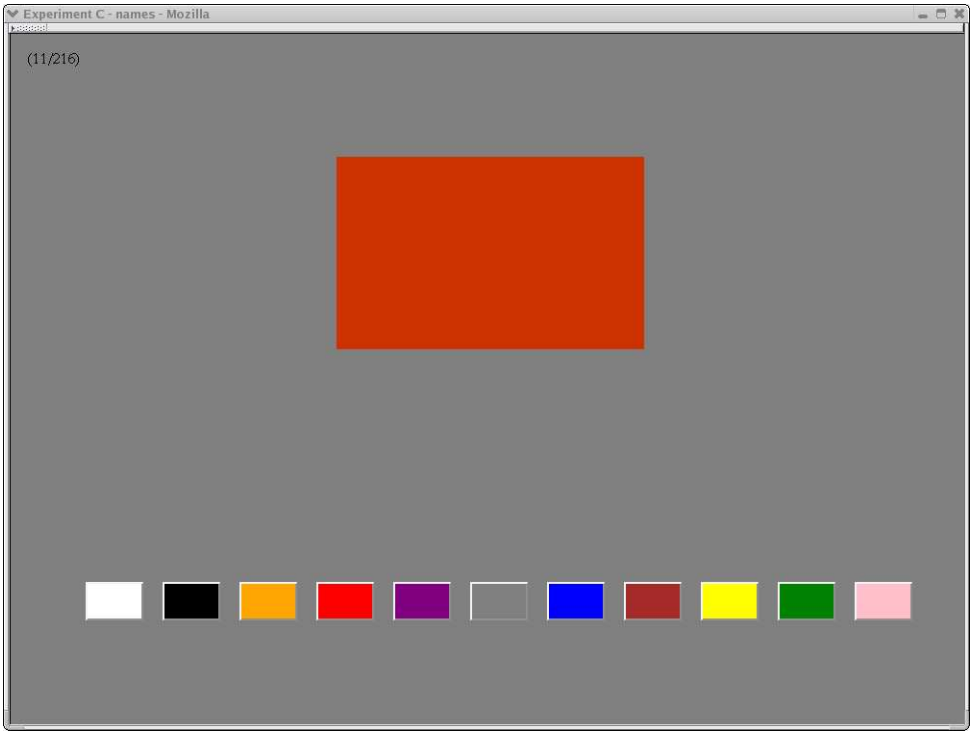
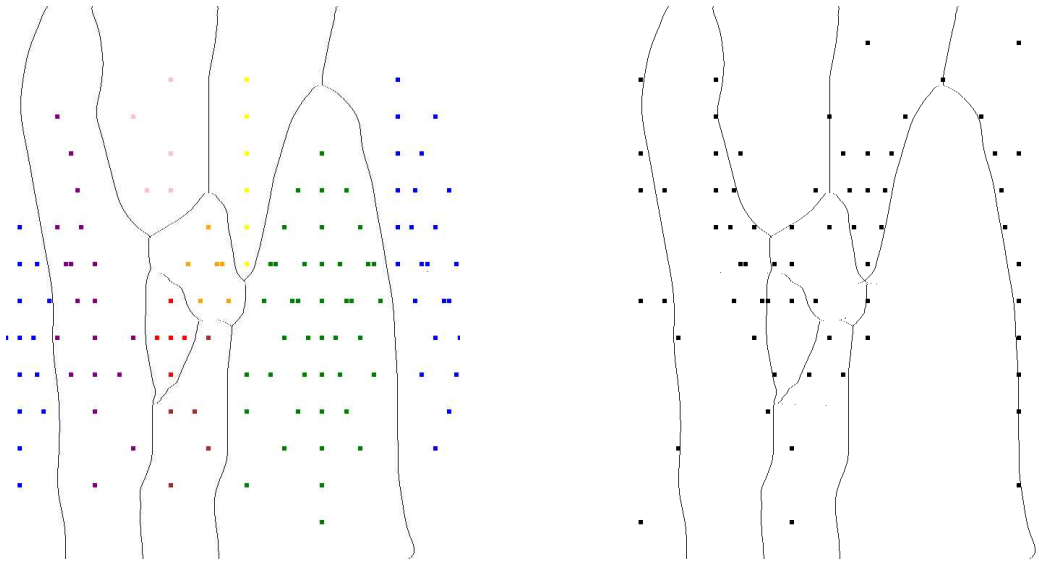
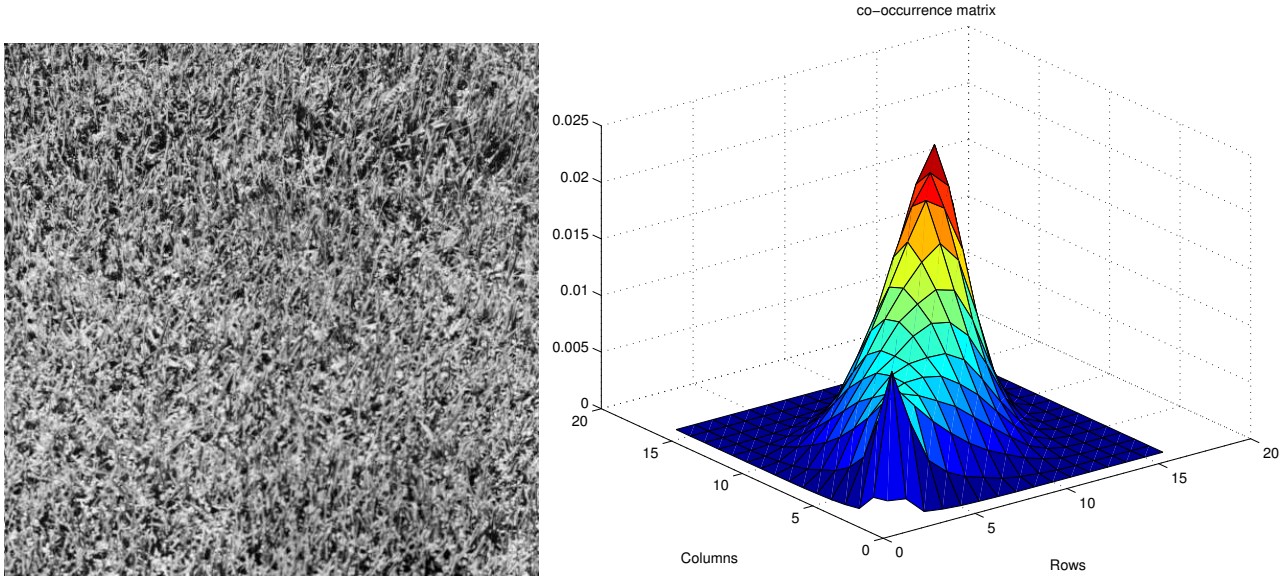


Figure B.3: Screendump of the user interface of the color discrimination experiment. The buttons were colored and did not have a label. Hence, the participants were able to compare the color of the stimulus with the colors of the buttons; a process of color discrimination.



(a) (b)

Figure B.4: The two dimensional HI plane with the calculated chromatic borders. (a) shows the non-fuzzy chromatic CLUT markers and (b) shows the fuzzy chromatic CLUT markers. Each dot represents a W3C web-safe color.



(a) (b)

Figure B.5: (a) shows an image from which a co-occurrence matrix and features are calculated. In (b) a visual rendering is plotted of the co-occurrence matrix of image (a).

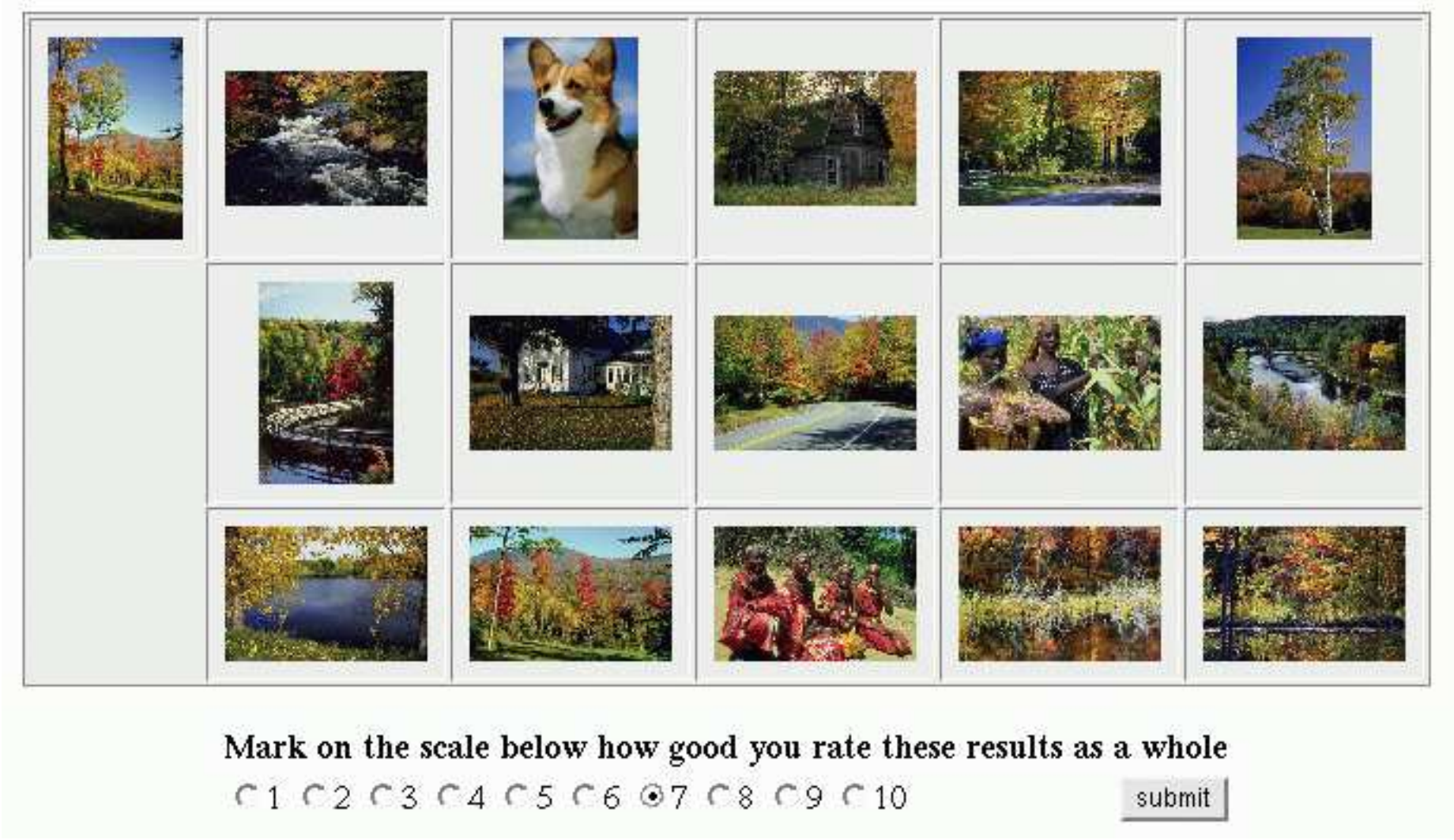


Figure B.6: The interface of a query such as was presented to the subjects. They were asked to select the best matching images and to rate their satisfaction.

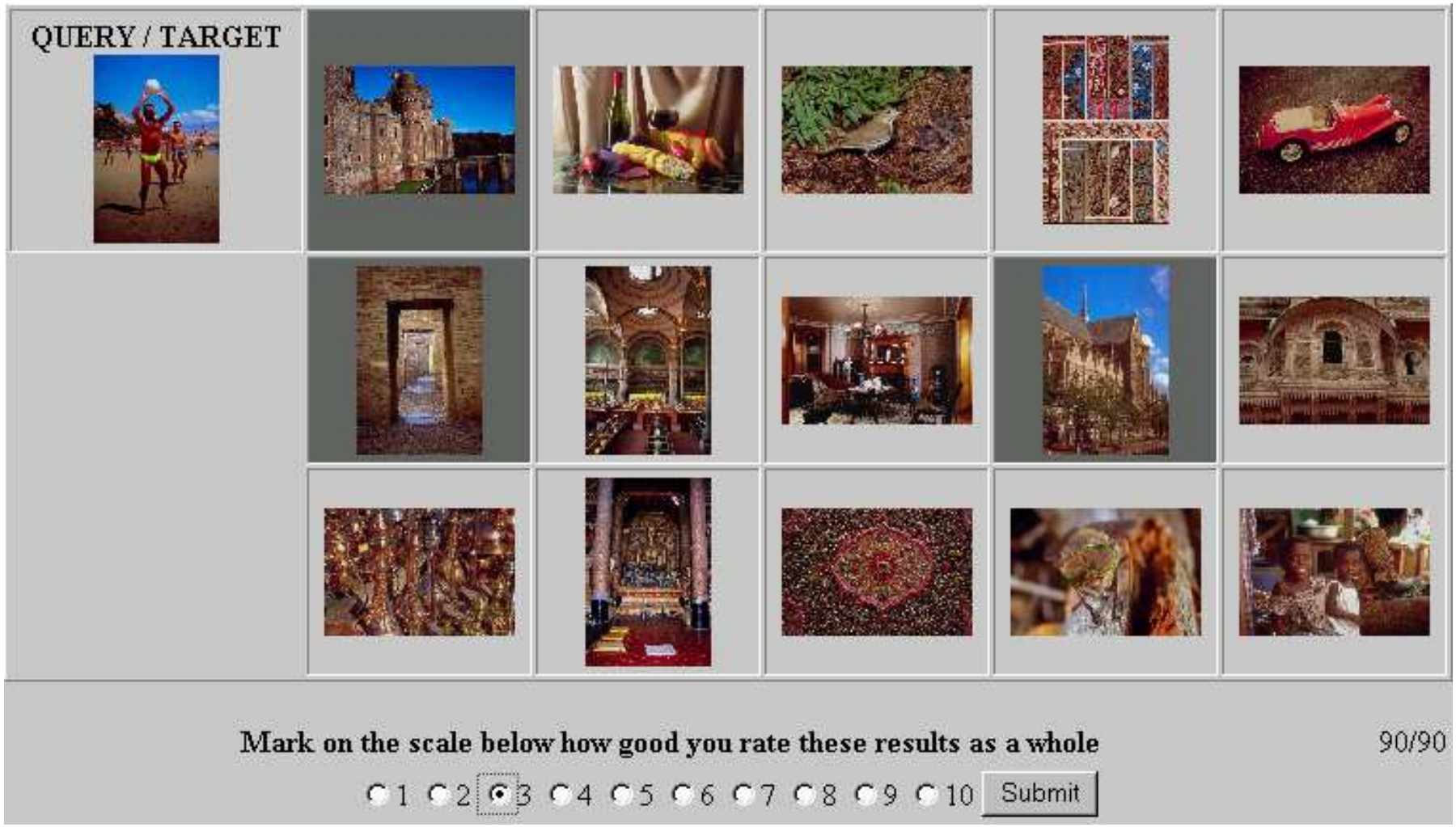


Figure B.7: The interface of a query as was presented to the participants. They were asked to select the best matching images and to rate their overall satisfaction, with respect to their color distribution only.

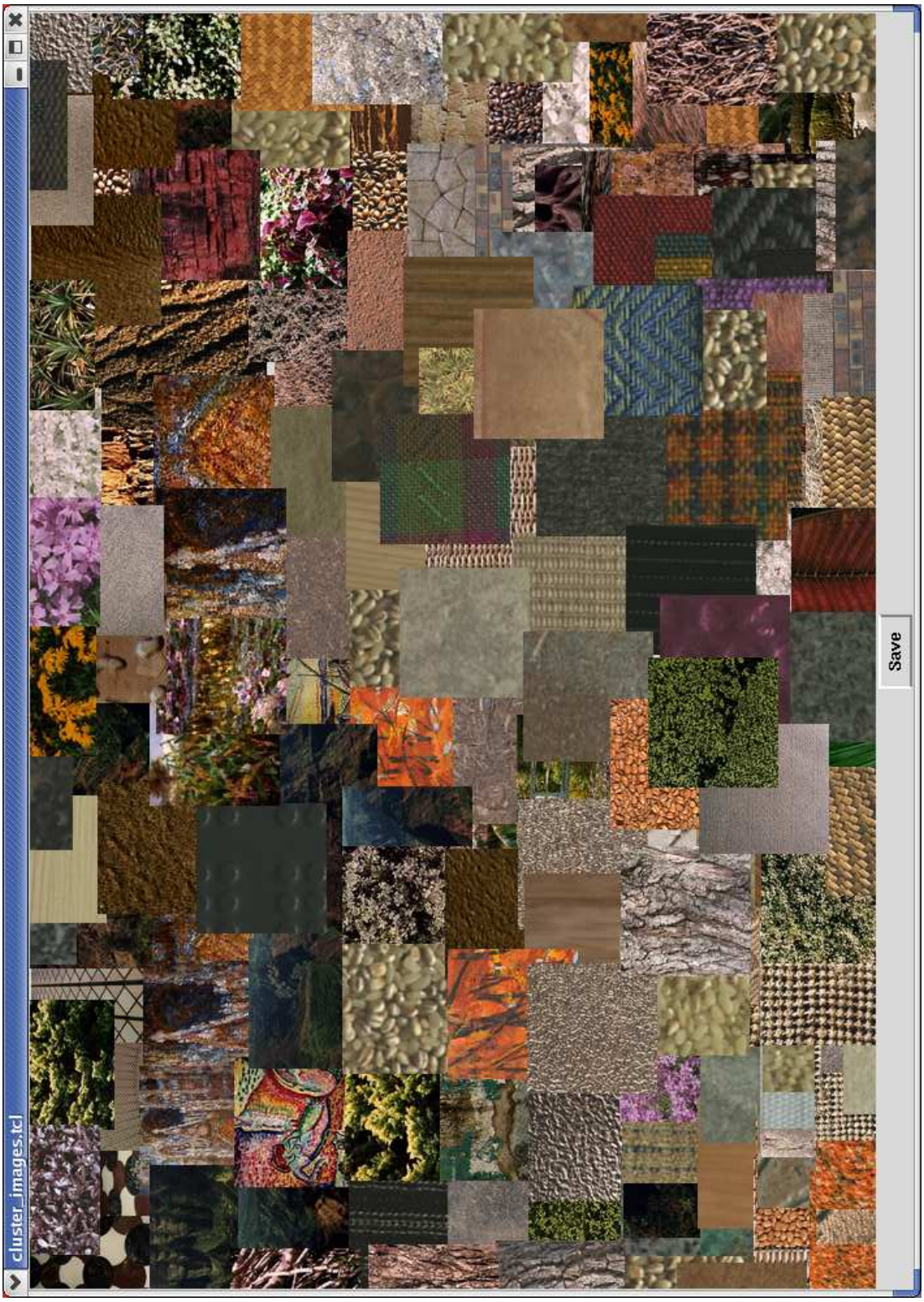


Figure B.8: An overview of all 180 images (the color version) used in the clustering experiments with both human participants and the automatic classifier.

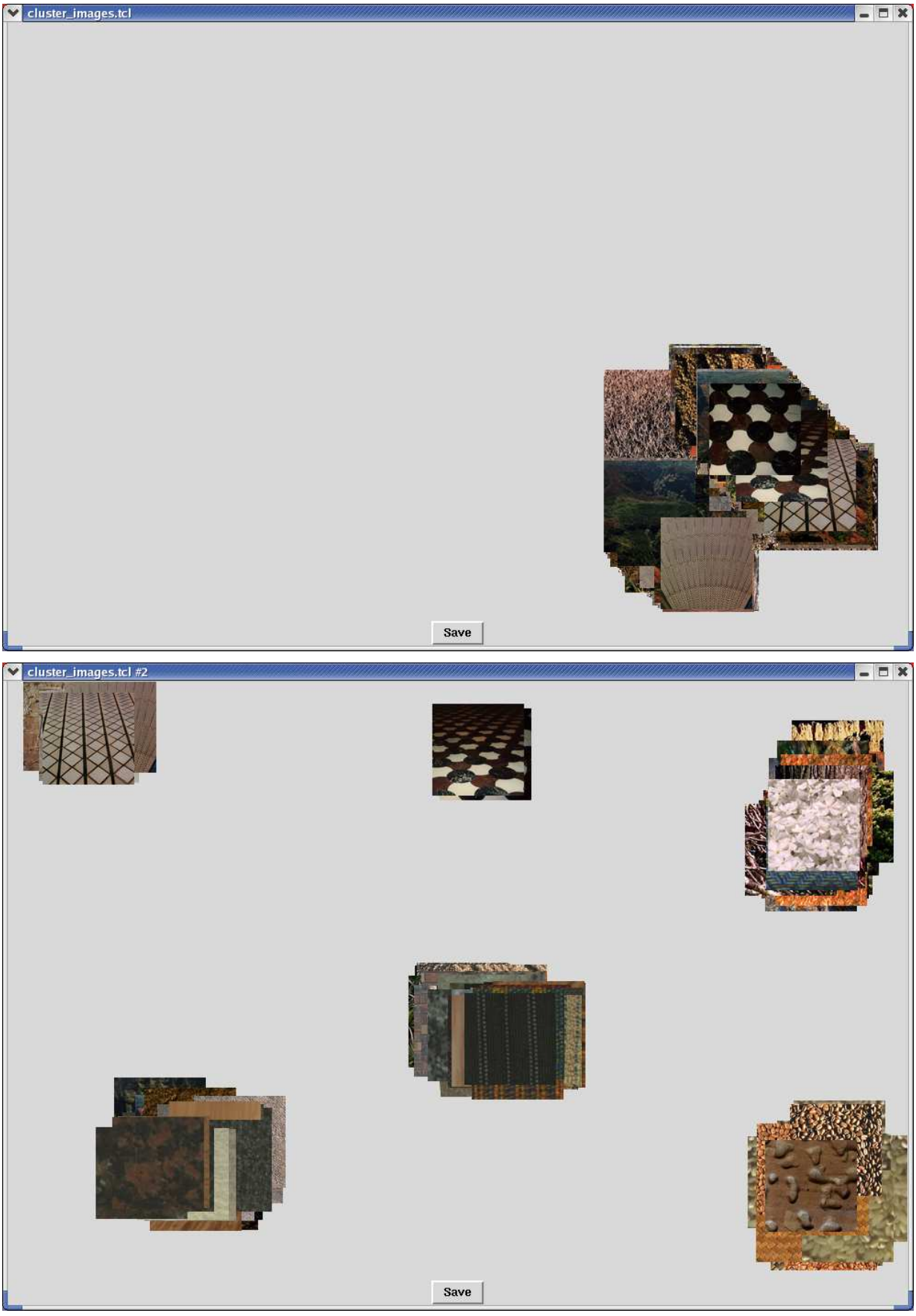


Figure B.9: Above: The start condition of the experiment: one pile of 180 images. Below: An example of a final result of an experiment: six clusters of images.

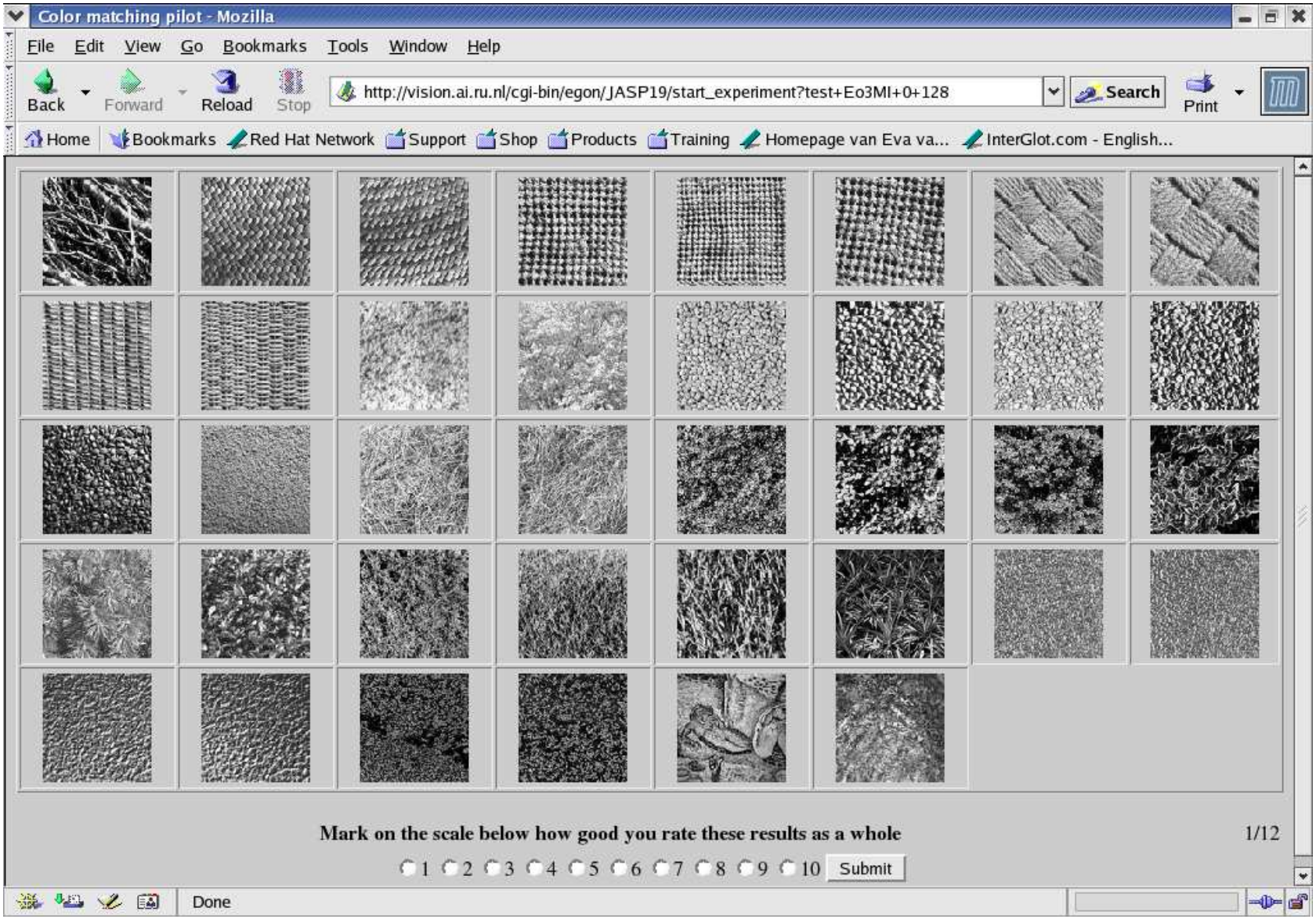


Figure B.10: An example screen from the benchmark used to let users judge the automatic clusters.

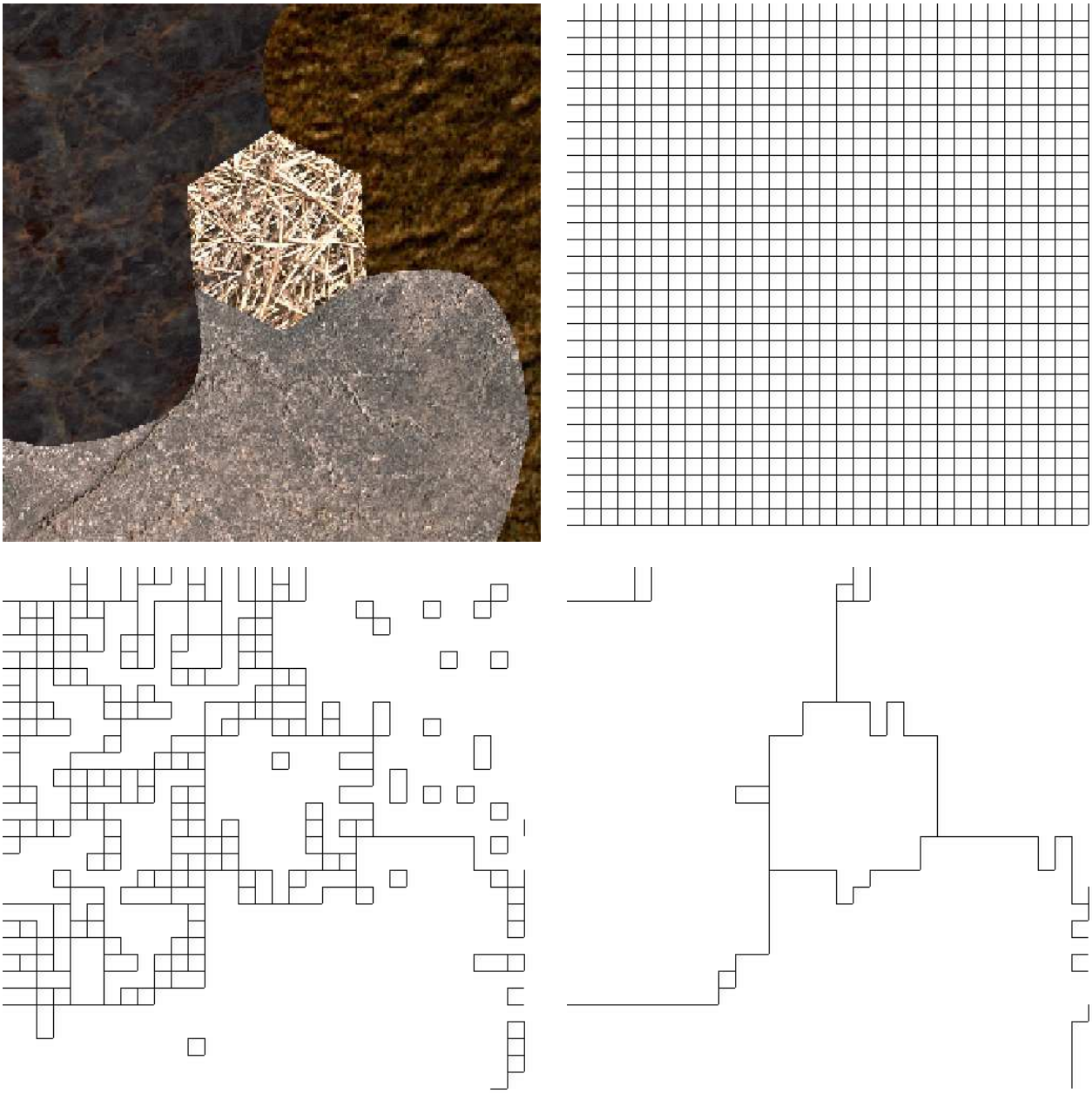


Figure B.11: The segmentation process, from left to right: The original image, division of the image in blocks of size 16×16, the regions after 800 iterations of agglomerative merging, and the final segments.

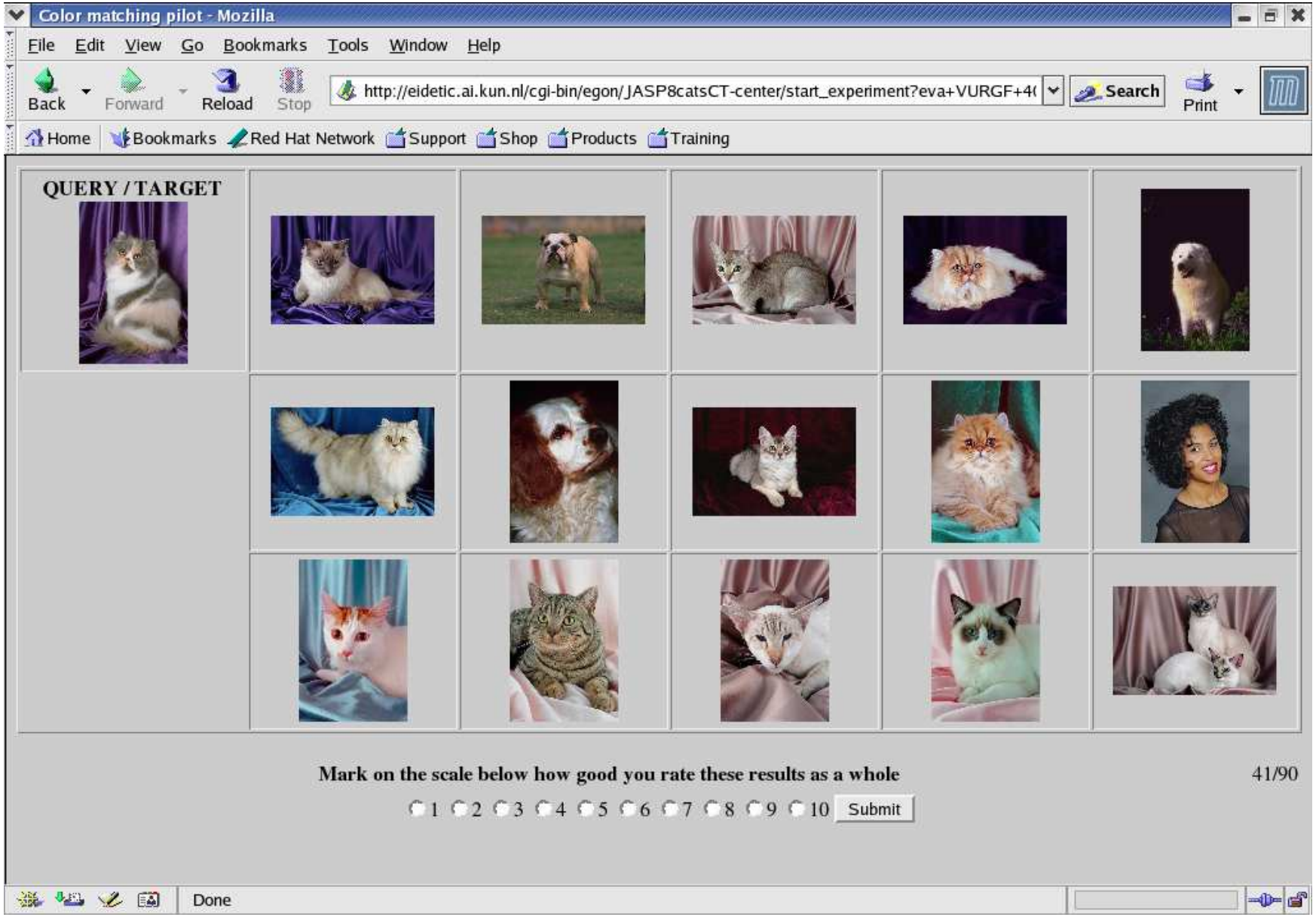


Figure B.12: A query image with retrieval results when using color and texture features for matching.

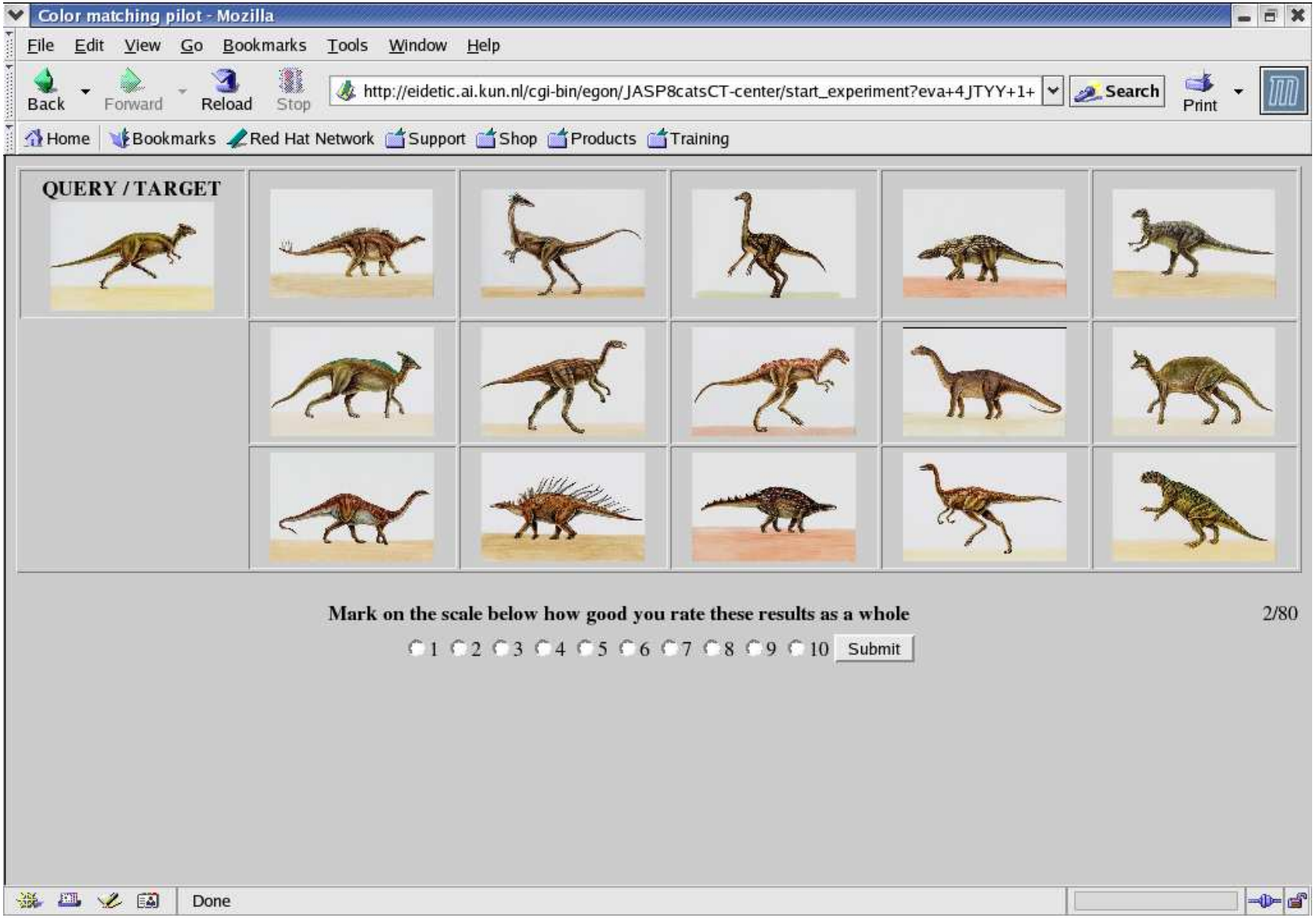
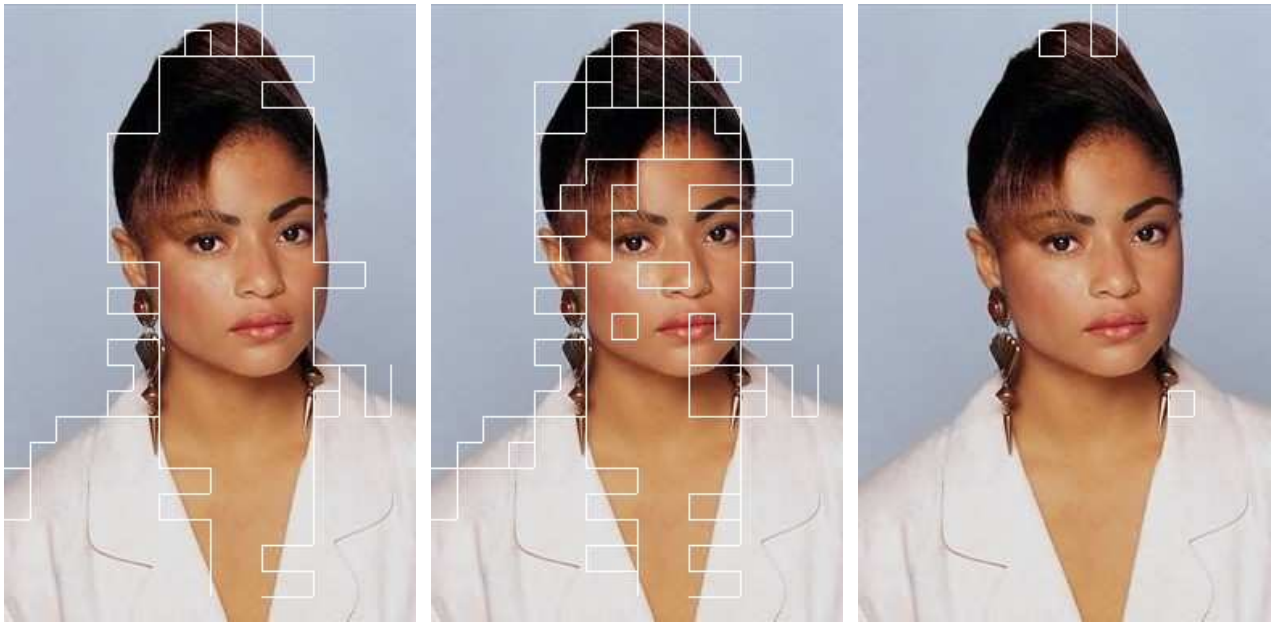
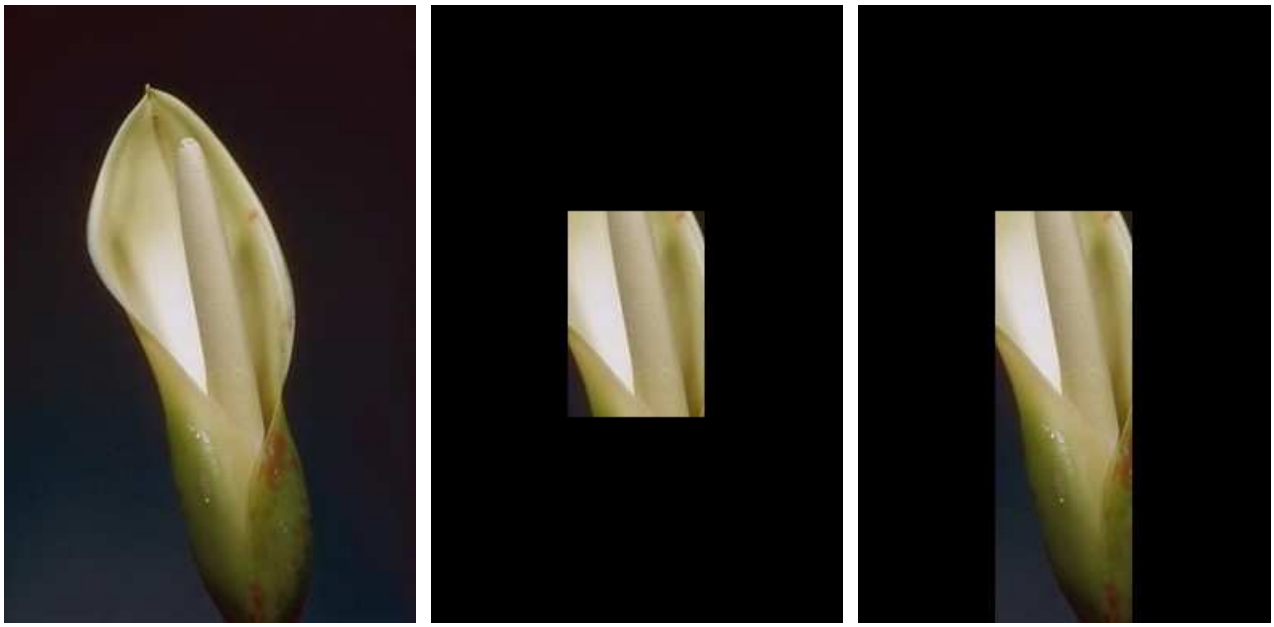


Figure B.13: A query image with retrieval results when using color and texture features for matching.



(a) (b) (c)

Figure B.14: Segmentation of images with several parameters: (a) The correct parameter for its class (0.700). (b) The generic parameter as used in phase 1 (0.600). (c) The parameter of the class cats (0.800).



(a) (b) (c)

Figure B.15: (a) The original image. (b) The $\frac{1}{9}$ center grid cell of the image as used for analysis. (c) The $\frac{2}{9}$ center grid cells of the image as used for analysis.

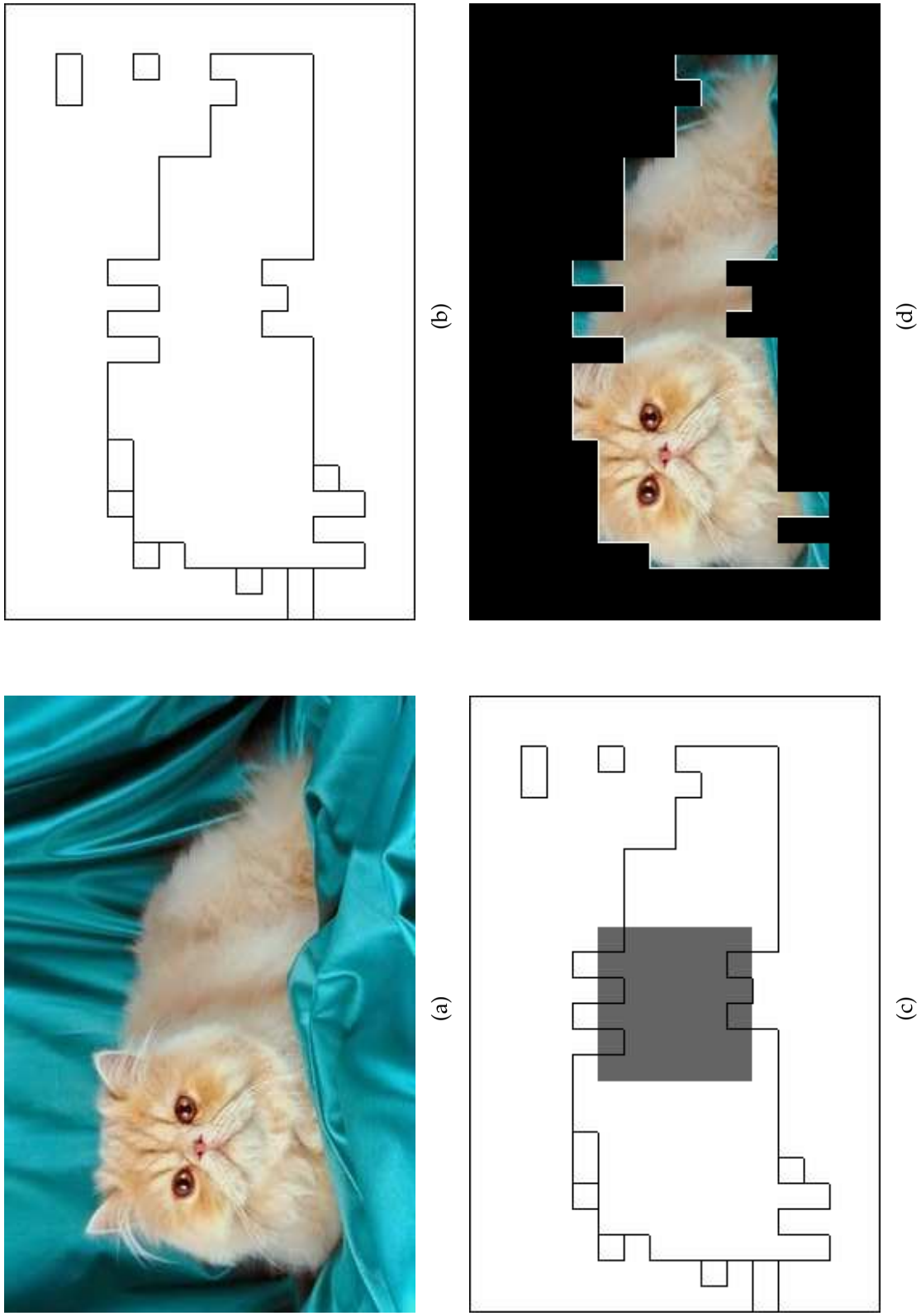


Figure B.16: (a) The original image. (b) The segments in the image (c) The grid. (d) The final region.

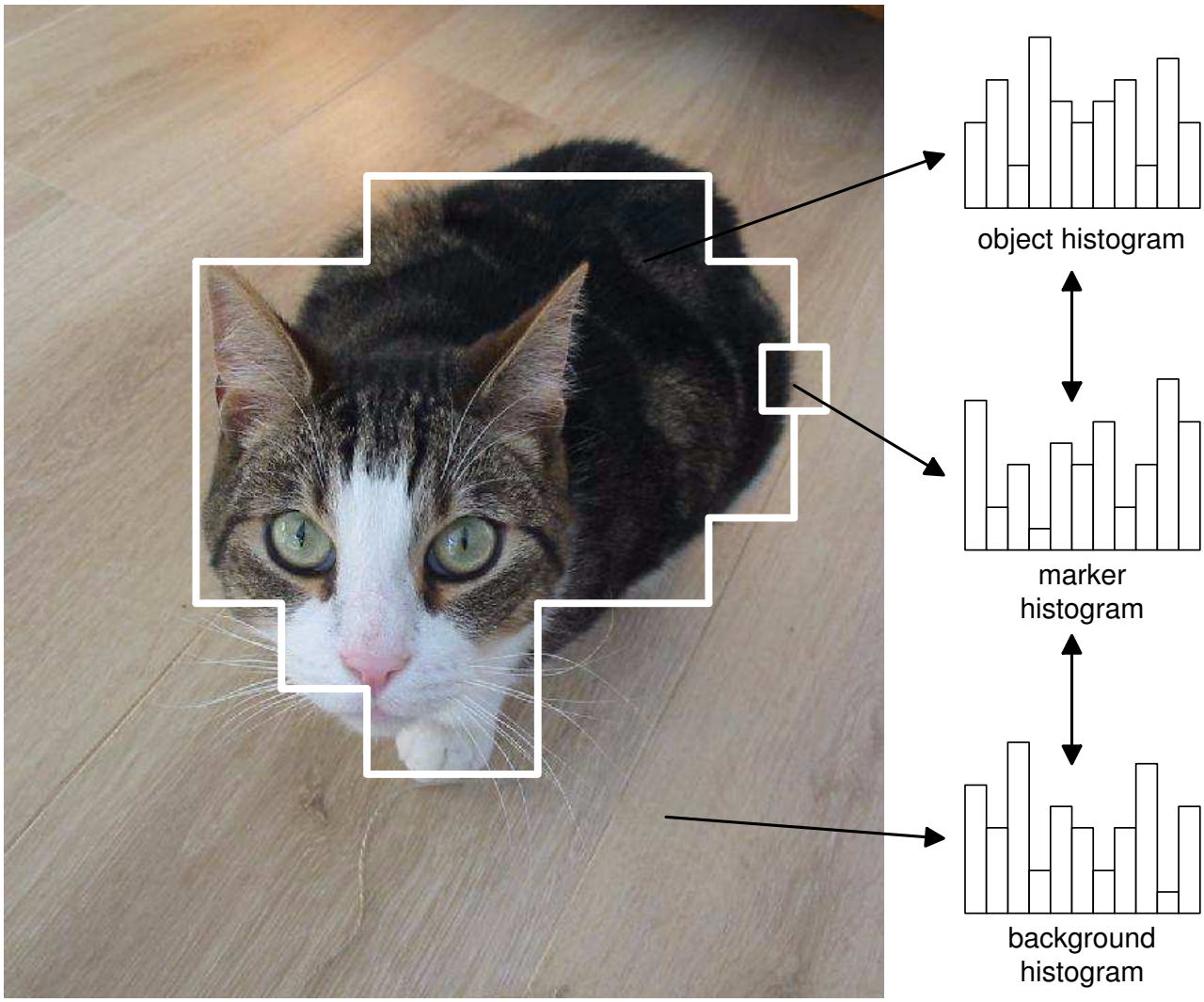


Figure B.17: The process of pixelwise classification illustrated. A pixel at the boundary is selected and a marker is placed over it. Next, the color histogram over this marker is calculated as well as the histograms of the center segment and the background. The histogram over the marker is compared to the other histograms and the pixel is assigned to the area with the most similar histogram (of the background or the object).

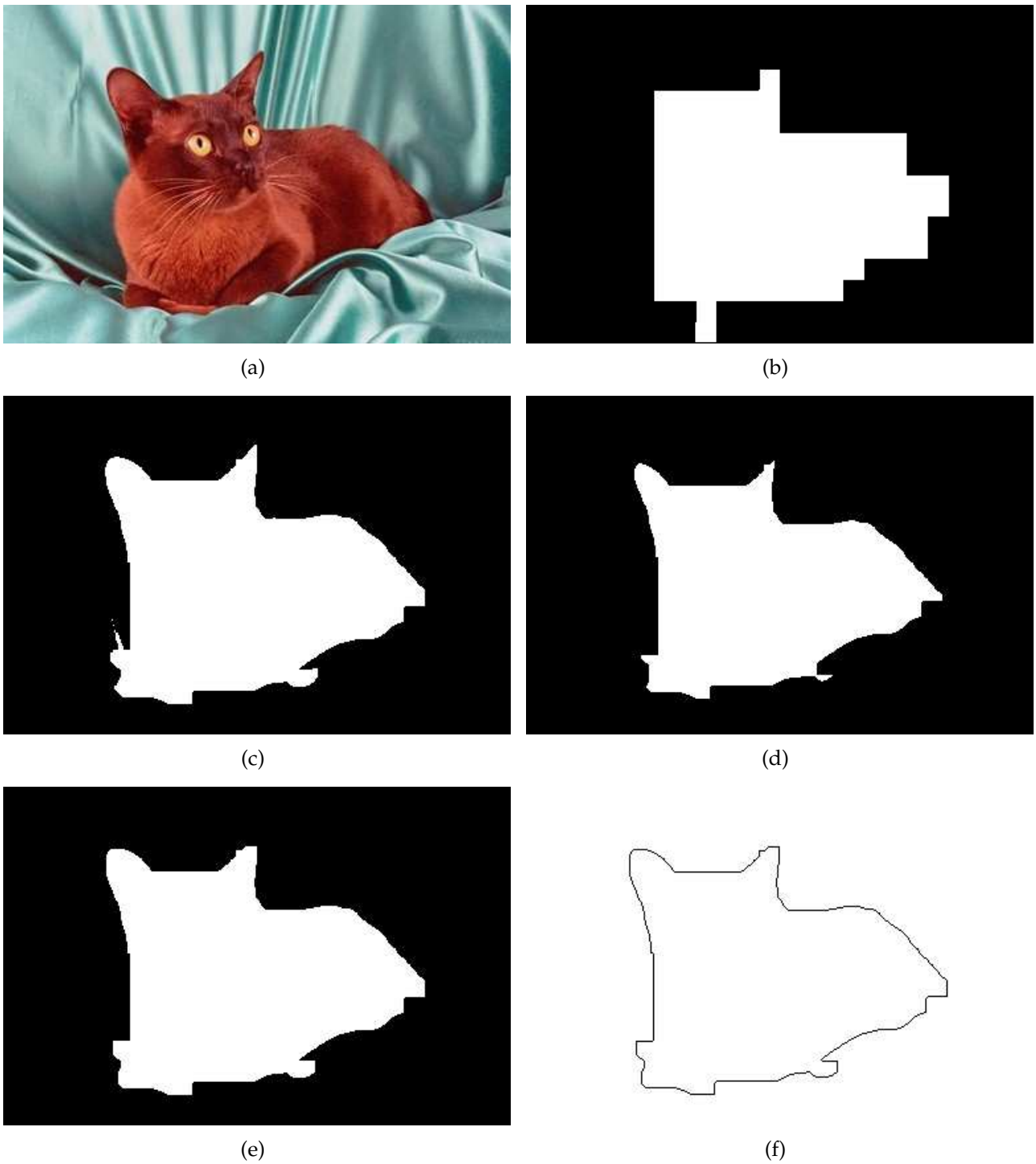
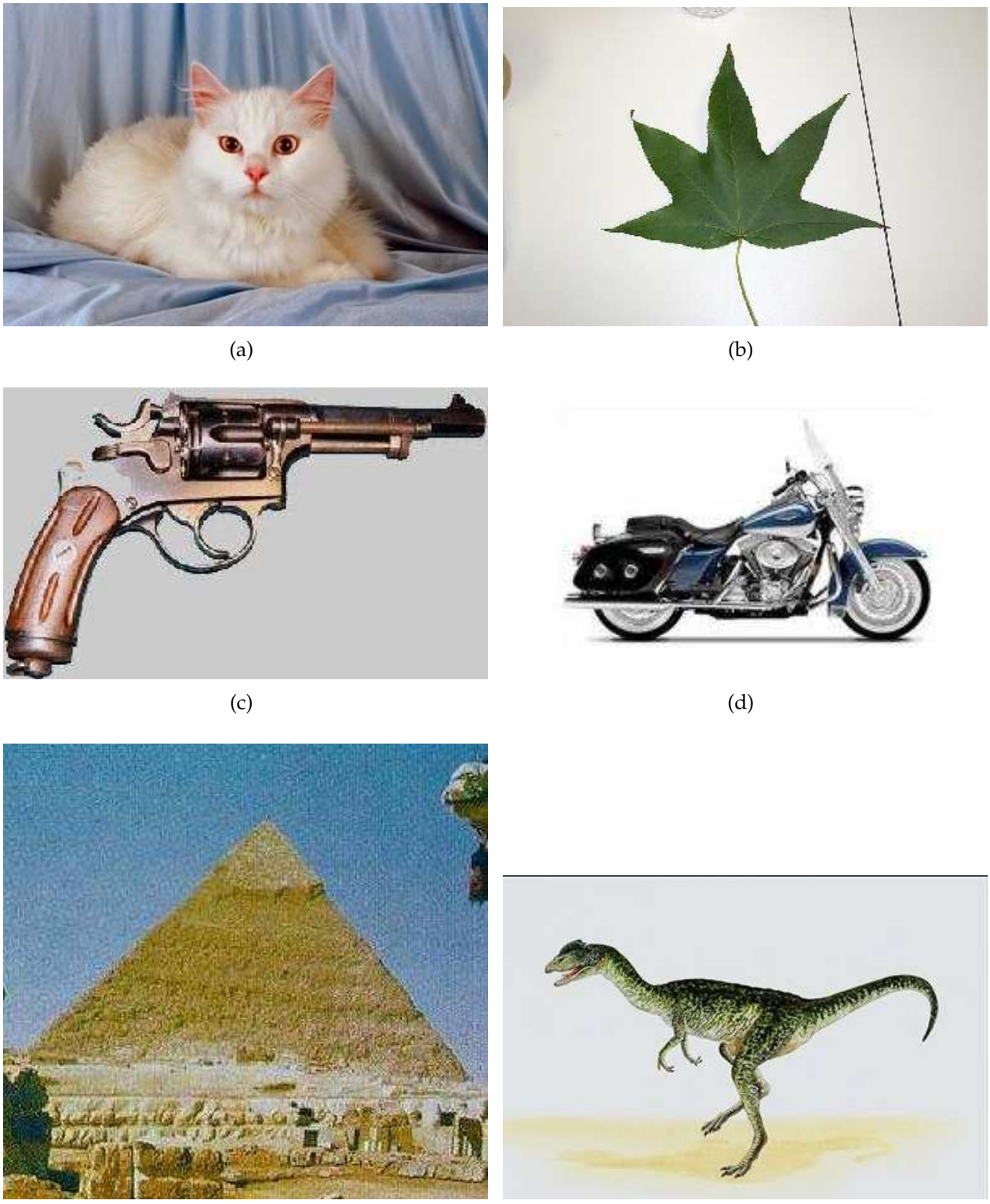


Figure B.18: (a) The original image (b) The coarse segmentation (c) The object after pixelwise classification (d) The object after erosion (e) The object after dilation (f) The final shape.



(a) (b)

(c) (d)

(e) (f)

Figure B.19: Sample images from the database used.

C

Fast Exact Euclidean Distance (FEED) transformation

Abstract

Fast Exact Euclidean Distance (FEED) transformation is introduced, starting from the inverse of the distance transformation. The prohibitive computational cost of a naive implementation of traditional Euclidean Distance Transformation, is tackled by three operations: restriction of both the number of object pixels and the number of background pixels taken in consideration and pre-computation of the Euclidean distance. Compared to the Shih and Liu 4-scan method the FEED algorithm is often faster and is less memory consuming.

This chapter is identical to:
Schouten, Th. E. and Broek, E. L. van den (2004). Fast Exact Euclidean Distance (FEED) Transformation. In J. Kittler, M. Petrou, and M. Nixon (Eds.), *Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR 2004)*, Vol 3, p. 594-597. August 23-26, Cambridge - United Kingdom.

C.1 Introduction

A distance transformation [233] (DT) makes an image in which the value of each pixel is its distance to the set of object pixels O in the original image:

$$D(p) = min\{dist(p,q), q \in O\} \tag{C.1}$$

Many algorithms to compute approximations of the Euclidean distance transformation (EDT) were proposed. Borgefors [21] proposed a chamfer DT using two raster scans on the image which produces a coarse approximation of the EDT. To get a result that is exact on most points but can produce small errors on some points, Danielsson [69] used four raster scans.

To obtain an exact EDT two step methods were proposed. Cuisenaire and Macq [65, 66] first calculated an approximate EDT using ordered propagation by bucket sorting. It produces a result similar to Danielsson's. Then, this approximation is improved by using neighborhoods of increasing size. Shih and Liu [258] started with four scans on the image, producing a result similar to Danielsson. A look-up table is then constructed containing all possible locations where an exact result is not produced. Because during the scans the location of the closest object pixel is stored for each image pixel, the look-up table can be used to correct the errors. It is claimed that the number of error locations is small.

In contrast with these approaches, we have implemented the EDT starting directly from the definition in Equation C.1. Or rather its inverse: each object pixel feeds its distance to all non-object pixel.

C.2 Direct application

In principle for each pixel q in the set of object pixels (O) the Euclidean distance (ED) must be calculated to each background pixel p. The algorithm then becomes:

$$\begin{aligned} &initialize\ D(p)\ =\ if\ (p \in O)\ then\ 0,\ else\ \infty \\ &\quad foreach\ q \in O \\ &\qquad foreach\ p \notin O \\ &\qquad\quad update:\ D(p) = min(D(p), ED(q,p)) \end{aligned}$$

However, this algorithm is extremely time consuming. In principle it can be speeded up by:

- restricting the number of object pixels that have to be considered
- pre-computation of $ED(q,p)$

- restricting the number of background pixels that have to be updated for each considered p

Only the "border" pixels of an object have to be considered. A border pixel is defined as an object pixel with at least one of its four 4-connected pixels in the background. It can then be easily proven that the minimal distance from any background pixel to an object, is the distance from that background pixel to a border pixel of that object.

As the ED is translation invariant, the EDs can be precomputed and stored in a matrix $M(x, y) = ED((x, y), 0)$. $ED(q, p)$ is then taken as $ED(q - p, 0)$ from the matrix. In principle the size of the matrix is twice the size of the image in each dimension. If the property $ED((x, y), 0) = ED((|x|, |y|), 0)$ is used in the updating of $D(p)$, only the positive quadrant of M is needed. Thus the size of the matrix becomes equal to the image size. Its calculation can be speeded up using the the fact that ED is symmetric: $ED((x, y), 0) = ED((y, x), 0)$.

If an upper limit of the maximum value of $D(p)$ in an image is known a priori, the size of M can be decreased to just contain that upper limit. This would increase the speed of the algorithm. For example, this could be done in a situation where there are fixed objects present. In addition, $D(p)$ can be calculated for the fixed objects only and can be updated in a later stage using only the pixels of the moving objects.

The size of the matrix M can also be decreased if one is only interested in distances up to a certain maximum. For example, in a robot navigation problem where distances above the maximum give no navigation limitations.

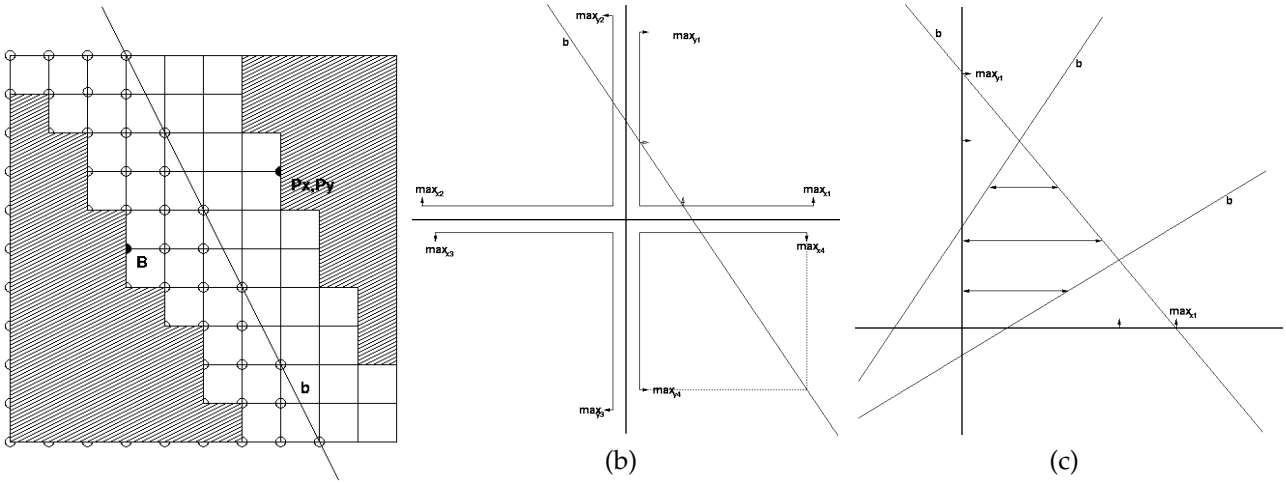
Due to the definition of $D(p)$ the matrix M can be filled with any non-decreasing function f of ED: $f(D(p)) = min(f(D(p)), f(ED(q, p)))$. For instance, the square of ED allowing the use of an integer matrix M in the calculation. Or one can truncate the ED to integer values in M if that is the format in which the final $D(p)$ is stored.

The number of background pixels that have to be updated can be limited: only those that have an equal or smaller distance to the border pixel B than to an object pixel p (see Figure C.1a). The equation of the bisection line is: $p_y y + p_x x = (p_x^2 + p_y^2)/2$.

Regarding the speed of the algorithm the problem is that not too much time should be spend on searching for other object pixels, on the administration of the bisection lines, or on determining which pixels to update. That is because the update operation is simply a test followed by one assignment, see the algorithm at the beginning of this section.

C.3 Reducing the number of pixels to update

The search for object pixels p is done on lines through the current border pixel (B) with certain $m = p_y/p_x$ ratio's. Define $m = m_1/m_2$ with m_1 and m_2 the minimal integers, then



(a) (b) (c)

Figure C.1: (a) Principle of limiting the number of background pixels to update. Only the pixels on and to the left of the line have to be updated. B is the border pixel under consideration, p is an object pixel. (b) Sizes of each update quadrant and changes due to a single bisection line and due to a crossing of two bisection lines. (c) The update process. On each scan line the bisection lines determine a range of pixels to update.

the equation of the bisection becomes: $2(m_1 m_2 y + m_2^2 x) = (m_1^2 + m_2^2)p_x$. This is of the form: $m_a y + m_b x = m_c p_x$ with m_a, m_b and m_c integers that depend only on m. For each quadrant for each m only the object pixel closest to B is relevant, searching along the line can be stopped as soon as one is found. The bisection line found is then identified with p_x and the quadrant number.

To keep track of the update area, the maximum x and y values of each quadrant are updated (see Figure C.1b). Only pixels inside each square need to be updated, but not all of them. A bisection line in a quadrant might update these maximum values in this and two neighboring quadrants, as is indicated with the open arrows in the figure. For example: $max_{y1} = min(max_{y1}, m_c p_x / m_a)$. The intersection point of two bisection lines in different quadrant might also give a new maximum value, as indicated with the arrow with the line on top. The maximum values can be calculated using integers.

The maximum values also determine the distance to search along each line m. For example, for a line in quadrant 1 at least one of the points (max_{x1}, max_{y1}), $(0, max_{y2})$, and $(max_{x4}, 0)$ must be on or to the right of the bisection line. This gives a maximum value for p_x of $max(m_a max_{y1} + m_b max_{x1}, m_a max_{y4}, m_b max_{x4})/m_c$.

Bisection lines closer to the origin B have a larger effect than lines further away. Searching in circular patterns the closest lines are found first, thus less points are checked than using a radial search. But it requires a more time consuming checking of reaching of limit values and further bookkeeping than when using a radial search.

Since in general a number of object points from the same object are close to B, the radial search is splitted. In a small area around B all the points on a number of radial lines are checked. If the remaining area ($\sum_{i=1}^{4} max_{xi} max_{yi}$) is too large, all the radial lines are checked completely. To increase the speed, not all points on them are checked but a certain stepping is used.

The final selection of pixels to update is made in the update process (see Figure C.1c). For each scan line in a quadrant bisection lines in that and neighboring quadrants determine start and end points, as is indicated with the lines with arrows on both sides. The maximum values in a quadrant can be at several locations, depending on whether crossing of bisection lines was taken into account. The crossing of a bisection line in a quadrant with a bisection line in the next quadrant is found when the minimum value of a scan line is larger than the maximum value. In that case no further scan lines in that quadrant have to be considered.

A further remark is that points exactly on a bisection line, have only to be updated once. For this quadrants 1 and 2 are chosen. This can simply be achieved by decreasing $m_c p_x$ by 1 for quadrants 3 and 4.

C.4 Test results

As comparison algorithm the four scan method of Shih [258] was chosen (except for their look-up table correction).

For our FEED method test lines with m's of 1/4, 1/3, 1/2, 2/3, 3/4, 1, 4/3, 3/2, 2, 3 and 4 and both the vertical and horizontal lines were chosen. The small test area around each border pixel was set to a square of 17 by 17 pixels and for the larger area a stepping of 8 pixels was used. This was skipped if the update area was already reduced to less than 200 pixels.

Results are presented on four 4096 by 4096 images and their negatives. Two are tests suggested by Cuisenaire and Macq [65, 66], an image containing a circle with a radius of 2000 pixels and one filled with a line under 20°. The other two images are filled with a large number of objects, one of them with situations in which errors in the implementation would likely show up.

Our implementation was checked on the small images given in reference [65, 66] and [258] that gives errors in ED in the first steps of the methods in the references, and was found to give correct results. The results on the test images using the algorithm in Section C.2 without any speed optimization, provided references for checking the correct implementation of the optimizations.

Table C.1: Timing results on eight images, for the Shih and Liu four scans method [258] and for the Fast Exact Euclidean Distance (FEED) transform.

image	Shih and Liu	FEED	reduced M
line	29.46 s	13.84 s	11.94 s
circle	22.86 s	17.42 s	15.77 s
test-obj	21.84 s	10.47 s	6.72 s
objects	23.39 s	8.15 s	5.52 s
neg line	5.26 s	5.01 s	4.07 s
neg circle	10.85 s	6.15 s	3.84 s
neg test-obj	9.96 s	7.14 s	4.00 s
neg objects	8.51 s	6.11 s	3.80 s

In Table C.1 the execution times on a SUN machine are given for the Shih and Liu method and for our FEED method with a full size matrix M and with the size of M reduced to the maximum distance in the image. FEED is up to a factor 3 faster then the Shih and Liu method and becomes even faster when the maximum distance in the image is known in advance.

Table C.2 shows some statistics gathered during the execution of our method. It shows the number of object and border pixels, the average number of tested pixels per border pixel and the average number of updates per background pixel. Together with Table C.1 it shows that although the execution time of the Shih and Liu method is proportional to the number of pixels in the image, in images of given size the execution time is about proportional to the number of background pixels. That is caused by the fact that during the four scans over the image, more work is performed on background pixels than on object pixels.

As noted by Cuisenaire and Macq [65, 66] comparing the complexity and computational costs of EDT algorithms is a complex task. FEED shows a factor 2 less variation in execution time over the images than the Shih and Liu method, which needs further investigation. Somehow, the work done in the various parts of FEED averages out better than in the Shih and Liu method.

Shih and Liu argue that the number of pixels with a wrong ED after their four scans over the image is very small, less than 1%. We found for the circle image that 8.32% and for the line image that 68.8% of the pixels were wrong. In the line under 20 $^\circ$ image the conditions under which the four-scans produce a wrong result, occur very often.

Table C.2: Image statistics: The number of tests done per border pixel and the number of updates done per background pixel.

image	object pixels	border pixels	tests / border pixel	updates / backgr pixel
line	16756554	4282	5010.6	8.86
circle	12579510	12770	2851.4	13.13
test-obj	13232644	115908	339.5	10.47
objects	14140711	74299	421.5	1.64
neg line	4282	8188	21.6	1.67
neg circle	418136	12766	430.0	4.20
neg test-obj	3528192	127728	110.4	1.48
neg objects	2620125	74586	131.7	1.63

C.5 Discussion

We have developed a Fast Exact Euclidean Distance (FEED) transformation, starting from the inverse of the distance transformation: each object pixel feeds its distance to all background pixels. The prohibitive computational cost of a naive implementation of traditional EDT, is tackled by three operations: restriction of both the number of object pixels and the number of background pixels taken into consideration and pre-computation of the Euclidean Distance.

The FEED algorithm was tested on 4 images and their negatives. It proved to be up to 3 times faster than the Shih and Liu 4-scan method. The difference is even larger if an upper bound on the maximum distance is known or if one is not interested in distances larger than a given maximum. In addition, was measured that the processing time of FEED is less variable than that of the 4-scan algorithm. Last, FEED needs less memory than the 4-scan method since it does not have to consider the image matrix twice.

Further research will focus on boosting FEED, design an parallel implementation (where each processor handles part of the border pixels, then joining of D(p)), and extend FEED to 3D or higher. So, with FEED no approximations of EDT are needed due to its computational burden, but both fast and exact EDT can be done. With that a new image processing algorithm is launched important for many applications in image analysis and pattern recognition.

Summary

This summary accompanied:

Broek, E. L. van den, Rikxoort E. M. van, Kisters, P. M. F., Schouten, Th. E., and Vuurpijl, L. G. (2005). Human-centered object-based image retrieval. In C. Klöditz (Ed.), *Proceedings of fourth NWO ToKeN symposium*, p. 5. March 18, The Netherlands - Eindhoven.

Digital media are rapidly replacing their analog counterparts. This development is accompanied by (i) the increasing amount of images present on the Internet, (ii) the availability of the Internet for an increasing number of people, (iii) a decline in digital storage costs, and (iv) the developments in personal digital video/photo camera's [30, 31, 126]. In anticipation of these developments, the fields of computer vision (CV) and content-based image retrieval (CBIR) evolved rapidly. Driven by a technology push, a range of CV/CBIR techniques were developed [125, 126]. However, seldomly the user and his characteristics were taken into account and subsequently, limitations of mere technical solutions became apparent [29, 30, 42].

The NWO ToKeN Eidetic project: Intelligent CBIR, aims to bridge the semantic gap present in the field of CV/CBIR, with the successful launch of the CBIR system Vind(X) as its foundation. However, the Vind(X) systems suffers from two drawbacks [38], it depends on: (i) the cooperative annotation of its users to fill its database of outlines [308] and on (ii) outline-outline (or shape) matching [38].

To enable a full analysis of image content (e.g., through object recognition), color and texture analysis has to be done as well as segmentation and shape extraction, to facilitate shape matching. Since each of these topics is essential for CV/CBIR, each of them was addressed in the Eidetic research line and will be discussed, before combining them.

Most images present on the Internet and in databases are color images. Moreover, the analysis of color in the image is not only used for the analysis of color distributions but is also used in texture analysis, image segmentation, and shape extraction. Hence, color analysis is of the utmost importance, for bridging the semantic gap [42], since color captures essential information about our environment [30, 31, 39, 225]. Therefore, we started with fundamental research toward human color processing [28]. This resulted in a unique color space segmentation, driven by experimental data concerning the 11 color categories, known to be used by humans since half a century [40, 41]. This color space segmentation can function as a highly efficient, human-based, color quantization scheme [30, 40].

Texture is the second feature, widely used for image analysis, CV, and CBIR purposes. Most texture analysis techniques are intensity-based [35, 36]. However, multiple problems can arise with texture analysis of color images, when their color is ignored (e.g., two distinct colors can have the same intensity). Therefore, intensity-based and color induced texture analysis were compared, using several color spaces and quantization schemes [35, 36]. This resulted in the new, parallel-sequential texture analysis approach with a 96% correct classification performance [36]. Moreover, in the research line "mimicking human texture classification", various texture analysis techniques were compared with human texture classification [39, 225].

Using the 11 color categories and the parallel-sequential texture analysis scheme, coarse image segmentation was conducted, the first phase of shape extraction [224]. Next, the exact shapes were extracted from such coarse segments by pixelwise classification, fol-

lowed by smoothing operators [38]. The shapes extracted were analyzed using the Vind(X) engine. With that, for all three features, human-based techniques have been developed to extract them from not annotated image material. Moreover, the segmentation and shape extraction enabled us to conduct object-based image retrieval (OBIR) [38].

In order to test the feature extraction techniques, an online CV/CBIR benchmark was developed [30, 31]. Retrieval was conducted utilizing: (i) color, (ii) texture, (iii) shape, (iv) color and texture combined, and (v) the three features combined. Object-based image retrieval, exploiting color, texture, and shape features, resulted in a retrieval precision up to 80% [38]. Hence, the first human-based, computationally very efficient, OBIR engine was launched.

In parallel to the research discussed so far, a research line in computational geometry emerged. Within this line of research the Fast Exact Euclidean Distance (FEED) transform was developed [254]. FEED was used in [40] for the color space segmentation and exploited for the Weighted Distance Mapping paradigm [41]. Recently, a parallel implementation of FEED: timed FEED was launched [253], is FEED applied for robot navigation [253] and video surveillance [252], and 3D-FEED was developed [251], accompanied by a dimension independent definition of the algorithm.

Samenvatting

This is a translation of the summary that accompanied:

Broek, E. L. van den, Rikxoort E. M. van, Kisters, P. M. F., Schouten, Th. E., and Vuurpijl, L. G. (2005). Human-centered object-based image retrieval. In C. Klöditz (Ed.), *Proceedings of fourth NWO ToKeN symposium*, p. 5. March 18, The Netherlands - Eindhoven.

Digitale media vervangen hun analoge tegenhangers in een hoog tempo door: (i) het toenemende aantal plaatjes op Internet, (ii) het gebruik van Internet door steeds meer mensen, (iii) het goedkoper worden van opslagruimte en (iv) de ontwikkelingen op het gebied van video en fotocamera's [30, 31, 126]. Zo werd het noodzakelijk te kunnen zoeken naar plaatjes, die mogelijk niet zijn beschreven, hetgeen content-based image retrieval (CBIR) wordt genoemd [125, 126]. Hierbij wordt gezocht met behulp van eigenschappen van de plaatjes zelf, in plaats van met hun beschrijvingen. Gedreven door de snelle technologische ontwikkelingen werd een verscheidenheid aan CBIR technieken ontwikkeld. Hierbij werden zelden de gebruikers of hun eigenschappen in ogenschouw genomen. Ten gevolge hiervan kwamen de beperkingen van puur technische oplossingen naar voren [29, 30, 42].

Het NWO ToKeN Eidetic project: Intelligent CBIR, heeft als doel CBIR technieken te ontwikkelen vanuit het perspectief van haar gebruikers. Eidetic werd geïnitieerd door de succesvolle lancering van het CBIR systeem Vind(X). Helaas was Vind(X) beperkt doordat ze afhankelijk is van de mede-werking van gebruikers om de database met omtrekken te vullen. Daarnaast worden door Vind(X) enkel omlijningen (of vormen) van objecten in plaatjes met elkaar vergeleken [38, 308].

Voor volledige analyse van het beeldmateriaal dient kleur en textuuranalyse te worden gedaan evenals segmentatie en vormextractie ten behoeve van het vergelijken van vormen van objecten. Daar ieder van deze onderwerpen essentieel zijn voor CBIR, zullen ze alle worden behandeld binnen de Eidetic onderzoekslijn. We zullen ieder van deze onderwerpen behandelen alvorens hen te combineren.

De meeste plaatjes op het Internet en in databases zijn kleurenplaatjes. Bovendien wordt kleuranalyse niet alleen gebruikt bij het bepalen van kleur distributies maar ook bij textuuranalyse, beeldsegmentatie en vormextractie. Een goede kleuranalyse is dus van groot belang [30, 31, 39, 42, 225]; daarom is een fundamenteel onderzoek naar menselijke kleurverwerking opgezet [28]. Dit resulteerde in een unieke kleurruimtesegmentatie, bepaald door de experimentele data betreffende de 11 kleur categorieën, zoals bekend sinds een halve eeuw [40, 41]. De kleurruimtesegmentatie kan dienen als een zeer efficiënt, op de mens gebaseerd schema voor kleurkwantisatie [30, 40].

Textuur is de tweede beeld eigenschap die veel gebruikt word bij CBIR doeleinden. Veel textuuranalysetechnieken zijn op intensiteit/grijswaardes gebaseerd [35, 36]. Indien kleur genegeerd wordt kunnen er diverse problemen naar voren komen bij textuuranalyse (bijvoorbeeld: twee verschillende kleuren kunnen dezelfde intensiteit hebben). Daarom hebben we textuuranalyse gebaseerd op intensiteit en op kleur, gebruikmakend van verschillende kleurruimtes en kwantisatie schema's, met elkaar vergeleken [35, 36]. Dit resulteerde in een nieuwe methode voor textuuranalyse: de parallel-sequentiële aanpak, die bewees in 96% van de gevallen textuur correct te classificeren [36]. Daarnaast werden diverse textuuranalysetechnieken vergeleken met menselijke textuurclassificatie [39, 225].

Gebruikmakend van de 11 kleur categorieën en de parallel-sequentiële textuuranalyse werd een groffe beeld segmentatie bewerkstelligd, de eerste fase in het proces van vorm extractie [224]. Op basis van deze groffe segmenten werden vervolgens de exacte vormen bepaald door de classificatie van pixels, gevolgd door het afvlakken van de vormen [38].

De geëxtraheerde vormen werden geanalyseerd met behulp van Vind(X). Daarmee zijn voor alle drie de beeldkenmerken, op de mens gebaseerde technieken ontwikkeld. De segmentatie en vorm extractie maken bovendien objectgebaseerd zoeken naar plaatjes mogelijk (object-based image retrieval (OBIR)) [38].

Een CBIR benchmark werd ontwikkeld om de nieuw ontwikkelde technieken te kunnen testen [30, 31]. In deze benchmark werden de beeld eigenschappen (i) kleur, (ii) textuur, (iii) vorm, (iv) de combinatie kleur en vorm evenals (v) de combinatie van kleur, textuur en vorm apart getest. Zo bleek dat OBIR, gebruikmakend van kleur, textuur en vorm, tot 80% correcte plaatjes vindt [38]. Hiermee is het eerste op de mens gebaseerde, computationeel zeer efficiënte, OBIR systeem gelanceerd.

Parallel aan het onderzoek besproken tot dusverre, is een onderzoekslijn in de computationele geometrie ontstaan. Binnen deze lijn van onderzoek is de "Fast Exact Euclidean Distance (FEED) transform" ontwikkeld [254]. FEED is gebruikt voor kleurruimtesegmentatie [40] en is toegepast voor het "Weighted Distance Mapping" paradigma [41]. Recent is een parallele implementatie van FEED ontwikkeld [253], is FEED toegepast op robotnavigatie [253] en videobewaking [252] en was 3D-FEED ontwikkeld [251] te samen met een dimensie onafhankelijke definitie van het algoritme.

Dankwoord

Nijmegen, 31 juli 2005

In 2000 (Amsterdam/Nijmegen) tijdens de organisatie van de IWFHR VII maakte Louis kennis met mij. Lambert verliet het NICI snel daarna en vertrok naar het hoge noorden. Louis nam zijn plaats over en bood mij een plaats als promovendus aan. Louis bedankt voor de kans, voor het vertrouwen dat je in me hebt gehad en vervolgens voor de vrijheid die je me hebt gegeven mijn eigen "pad der promotie" te bewandelen.

Begin 2003 begon ik met een cursus beeldverwerking. Theo, bedankt voor alles wat ik tijdens en na de cursus van je heb mogen leren maar bovenal bedankt voor de prettige en buitengewoon vruchtbare samenwerking die als vanzelf daaruit volgde. De kennis opgedaan tijdens mijn studie te samen met hetgeen dat ik van jou heb geleerd hebben de belangrijkste bouwstenen gevormd voor dit proefschrift.

Charles, als "low profile" promotor, heb je me vaak terloops enkele tips gegeven gedurende mijn promotie. Stuk voor stuk bleken deze van grote waarde. In de laatste fase van de promotie heb je het manuscript van het proefschrift buitengewoon grondig doorgenomen en vervolgens aangepaste delen nogmaals van commentaar voorzien. Dit deed je bovendien keer op keer zo snel dat ik soms niet begreep waar je de tijd vandaan toverde. Hartstikke bedankt hiervoor!

Peter, het is allemaal hard gegaan: van studiegenoot, mede bestuurslid van CognAC en vriend tot student van me. Na enkele vakken bij me te hebben gedaan, ben je gestart met je afstuderen bij me. Zo ben je gedurende het grootste deel van mijn promotieproject er op alle mogelijke manieren intensief bij betrokken geweest. De basis voor dit proefschrift heb ik samen met jou gelegd.

Maarten, als 1e jaars deed je een bijzonder practicum functieleer–neurale netwerken bij me. Twee jaar later heb je, in het kader van een aantal vakken en zonder het op dat moment te weten, meegewerkt aan de experimenten die de daadwerkelijke start betekende van het onderzoek dat in dit proefschrift is beschreven.

Thijs, jij kwam, een paar jaar geleden alweer, met Maarten mee. Je hebt een aantal vakken en je B.Sc. afstudeerwerk bij me gedaan. Inmiddels ben je bezig met je M.Sc. afstudeerwerk. Helaas is dit werk gesneuveld bij de selectie voor dit proefschrift. Maar met onder andere de projecten C-BAR/M4ART en SUIT/C-SUIT ben je van bijna het begin tot het einde betrokken geweest bij het promotieproject.

Eva tijdens een borrel hebben we het over een mogelijke stage voor je gehad. Zo kwam het dat je vanaf eind 2003 tot begin 2005 hard hebt meegewerkt aan diverse experimenten die in dit proefschrift beschreven staan. Zelfs al had ik even stil willen blijven staan ik had van jou, zo vermoed ik, de tijd niet gekregen. Met een projectvoorstel dat ik klaar had liggen

bij de start van je afstuderen en met een duidelijke onderzoekslijn en planning voor ogen is het erg hard gegaan met je afstuderen en zo heb je meegewerkt aan een substantieel deel van mijn promotieproject.

Menno, min of meer toevallig kwam je tijdens m'n promotieproject tussendoor "zeilen". Af en toe was er een afspraak inzake je scriptie of voor het schrijven van een artikel. In begin mei 2005, rondom de dag van het Levenslied, heb jij in één week je scriptie afgemaakt en heb ik m'n manuscript afgemaakt. Een zeldzaam intensieve week waar ik een bijna surrealistisch "op kamp gevoel" aan over heb gehouden.

Peter, Maarten, Thijs, Eva en Menno, wat jullie gemeen hebben is het enthousiasme voor het onderzoek dat jullie deden. Daarbij was het nooit een probleem om een paar uurtjes in de avond of in het weekend "door te trekken". Het heeft bij jullie allemaal tot ongelofelijke resultaten geleid. Jullie hebben, zonder het te weten, mijn promotie tot een succes gemaakt. Naast jullie inhoudelijke bijdrage is het vooral de begeleiding van / het samenwerken met jullie geweest die mijn promotie de voor mij noodzakelijke dimensie van "mensenwerk" gaf. Zonder jullie was de wetenschap mij wellicht te eenzaam geweest. Maar ook alle andere studenten (ik ga jullie niet allemaal opnoemen), die ik les heb gegeven of heb begeleid bij een practicum of afstuderen; bedankt!

Iedereen van CogW bedankt! Ik heb altijd een 'thuisgevoel' bij CogW gehad, een betrokkenheid die aan de basis stond voor een prettige studeer- en werksfeer. In het bijzonder wil ik drie mensen bedanken. Eduard en Ton, mijn waardering voor jullie beiden laat zich niet in enkele woorden vatten, dus daar begin ik ook maar niet aan. Merijn, ik denk dat wij in veel opzichten elkaars tegenpolen zijn, des te verrassender is het eigenlijk dat we het zo goed met elkaar konden vinden als kamergenoten.

Vrienden, wat heb ik jullie verwaarloosd de afgelopen jaren. Herhaaldelijk heb ik beterschap beloofd, steeds weer lukte het niet. Er is te veel in het leven dat ik wil doen. Het blijft moeilijk te accepteren dat er fysieke en psychische grenzen zijn, maar boven alles is het chronische tekort aan tijd een probleem. Voor mij waren jullie dan wel regelmatig uit het oog maar nimmer uit het hart. Jullie waren mijn bakens op momenten dat het leven voorbij schoot op een onnavolgbaar traject, met een veel te hoge snelheid.

Lieve pa, ma en zusje, met jullie als steun is geen berg te hoog of dal te diep of zij kan overwonnen worden. Oneindig veel dank voor alles!

Egon

Publications

Articles and bookchapters

30. Broek, E. L. van den, Kok, T., Schouten, Th. E., and Hoenkamp, E. (2005). Online Multimedia Access (OMA): Two dimensions of freedom for laypersons and experts. *[submitted]*

29. M. Israël and E. L. van den Broek and P. van der Putten. VICAR VISION: A Multi-stage processing paradigm for content-based video retrieval. In V. A. Petrushin and L. Khan (Eds.), *Multimedia Data mining and Knowledge Discovery (Part III, Chapter 12).* Springer-Verlag: Berlin - Heidelberg. *[invited contribution]*

28. Broek, E. L. van den, Kok, T., Schouten, Th. E., and Hoenkamp, E. (2005). Multimedia for Art ReTrieval (M4ART). *[submitted]*

27. Schouten, Th. E., Kuppens, H. C., and Broek, E. L. van den. Video surveillance using distance maps. *[submitted]*

26. Broek, E. L. van den, Rikxoort, E. M. van, and Schouten, Th. E.. An exploration in modeling human texture recognition. *[submitted]*

25. Schouten, Th. E., Kuppens, H. C., and Broek, E. L. van den. Three Dimensional Fast Exact Euclidean Distance (3D-FEED) Maps. *[submitted]*

24. Geuze, J. and Broek, E. L. van den. Intelligent Tutoring Agent for Settlers of Catan. *[submitted]*

23. Broek, E. L. van den, Schouten, Th. E., and Kisters, P. M. F. Efficient Human-centered Color Space Segmentation. *[submitted]* .

22. Broek, E. L. van den, Kok, T., Hoenkamp, E., Schouten, Th. E., Petiet, P. J., and Vuurpijl, L. G. (2005). Content-Based Art Retrieval (C-BAR). In ... (Eds.), *Proceedings of the XVIth International Conference of the Association for History and Computing* , p. ...–... . September 14–17, The Netherlands – Amsterdam. *[in press]*

21. Broek, E. L. van den and Rikxoort, E. M. van (2005). Parallel-Sequential Texture Analysis. *Lecture Notes in Computer Science (Advances in Pattern Recognition) , 3687* , 532–541.

20. Broek, E. L. van den, Rikxoort, E. M. van, and Schouten, Th. E. (2005). Human-centered object-based image retrieval. *Lecture Notes in Computer Science (Advances in Pattern Recognition) , 3687* , 492–501.

19. Doesburg, W. A. van, Heuvelink, A., and Broek, E. L. van den (2005). TACOP: A cognitive agent for a naval training simulation environment. In M. Pechoucek, D. Steiner, and S. Thompson (Eds.), *Proceedings of the Industry Track of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-05)* , p. 35–42. July 25–29, Utrecht – The Netherlands.

18. Broek, E. L. van den (2005). Empathic Agent Technology (EAT). In L. Johnson, D. Richards, E. Sklar, and U. Wilensky (Eds.), *Proceedings of the AAMAS-05 Agent-Based Systems for Human Learning (ABSHL) workshop* , p. 59-67. July 26, Utrecht - The Netherlands.

17. Broek, E. L. van den, Jonker, C. M., Sharpanskykh, A., Treur, J., and Yolum, p. (2005). Formal modeling and analysis of organizations. In O. Boissier, V. Dignum, E. Matson, and J. S. Sichman (Eds.), *Proceedings of the AAMAS-05 Workshop From Organizations to Organization Oriented Programming (OOOP2005)* , p. 17–32. July 26, Utrecht - The Netherlands.*

16. Doesburg, W. A. van, Heuvelink, A., and Broek, E. L. van den (2005). TACOP: A cognitive agent for a naval training simulation environment. In F. Dignum, V. Dignum, S. Koendig, S. Kraus, M. P. Singh, and M. Wooldridge (Eds.), Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-05), vol. 3, p. 1363-1364. July 25-29, Utrecht - The Netherlands.

15. Rikxoort, E. M. van, Broek, E. L. van den, and Schouten, Th. E. (2005). Object based image retrieval: Utilizing color and texture. In B. J. A. Kröse, H. J. Bos, E. A. Hendriks, and J. W. J. Heijnsdijk (Eds.), *Proceedings of the Eleventh Annual Conference of the Advanced School for Computing and Imaging* , p. 401–408. June 8–10, The Netherlands – Heijen.

14. Broek, E. L. van den, Kisters, P. M. F., and Vuurpijl, L. G. (2005). Content-Based Image Retrieval Benchmarking: Utilizing Color Categories and Color Distributions. *Journal of Imaging Science and Technology , 49(3)* , 293–301.

13. Broek, E. L. van den, Schouten, Th. E., Kisters, P.M.F., and Kuppens H.C. (2005). Weighted Distance Mapping (WDM). In N. Canagarajah, A. Chalmers, F. Deravi, S. Gibson, P. Hobson, M. Mirmehdi, and S. Marshall (Eds.), *Proceedings of The IEE International Conference on Visual Information Engineering (VIE2005)* , p. 157–164. April 4–6, Glasgow – United Kingdom. Wrightsons - Earls Barton, Northants, Great Britain.

12. Schouten, Th. E., Kuppens, H. C., and Broek, E. L. van den (2005). Timed Fast Exact Euclidean Distance (tFEED) Maps. *Proceedings of SPIE (Real-Time Imaging IX), 5671* , 52–63.

11. Rikxoort, E. M., Broek, E. L. van den, and Schouten, Th. E. (2005). Mimicking human texture classification. *Proceedings of SPIE (Human Vision and Electronic Imaging X), 5666* , 215–226.

10. Broek, E. L. van den and Rikxoort, E. M. van (2004). Evaluation of color representation for texture analysis. In R. Verbrugge and L.R.B. Schomaker and N. Taatgen (Eds.), *Proceedings of the Belgian Dutch Artificial Intelligence Conference (BNAIC) 2004* , p. 35–42.

*Since all authors did have an equal contribution to the paper, an alphabetic listing of authors is chosen.

October 21–22, Groningen – The Netherlands.

9. Schouten, Th. E. and Broek, E. L. van den (2004). Fast Exact Euclidean Distance (FEED) Transformation. In J. Kittler, M. Petrou, and M. Nixon (Eds.), *Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR 2004)* , Vol. 3, p. 594–597. August 23–26, Cambridge – United Kingdom.

8. Israël, M., Broek, E. L. van den, Putten, P. van der, and Uyl, M. J. den (2004). Automating the construction of scene classifiers for content-based video retrieval. In L. Khan and V.A. Petrushin (Eds.), *Proceeding of the Fifth International Workshop on Multimedia Data Mining (MDM/KDD'04)* , p. 38–47. August 22, Seattle, WA – USA.

7. Broek, E. L. van den (2004). Emotional Prosody Measurement (EPM): A voice-based evaluation method for psychological therapy effectiveness. In L. Bos, S. Laxminarayan, and A. Marsh (Eds.), *Medical and Care Compunetics 1 (pp. 118–125).* Amsterdam, The Netherlands: IOS Press.

6. Broek, E. L. van den, Kisters, P. M. F., and Vuurpijl, L. G. (2004). Design Guidelines for a Content-Based Image Retrieval Color-Selection Interface. In B. Eggen, G. van der Veer, and R. Willems (Eds.), *Proceedings of the ACM SIGCHI.NL 2004 conference: Dutch Directions in HCI.* June 10, The Netherlands – Amsterdam.

5. Broek, E. L. van den, Kisters, P. M. F., and Vuurpijl, L. G. (2004). The utilization of human color categorization for content-based image retrieval. *Proceedings of SPIE (Human Vision and Electronic Imaging IX), 5292* , 351–362.

4. Broek, E. L. van den, Hendriks, M. A., Puts, M. J. H., and Vuurpijl, L. G. (2003). Modeling human color categorization: Color discrimination and color memory. In T. Heskes, P. Lucas, L. G. Vuurpijl, and W. Wiegerinck (Eds.), *Proceedings of the Belgian Dutch Artificial Intelligence Conference (BNAIC2003)* , p. 59–66. October 23–24, The Netherlands – Nijmegen.

3. Broek, E. L. van den (2003). A stress marker in speech. *Toegepaste Taalwetenschap in Artikelen , 69* , 143–153 & 172.

2. Broek, E. L. van den, Vuurpijl, L. G., Kisters, P. M. F., and Schmid, J. C. M. von (2002). Content-Based Image Retrieval: Color-selection exploited. In M.-F. Moens, R. De Brusser, D. Hiemstra, and W. Kraaij (Eds.), *Proceedings of the 3rd Dutch-Belgian Information Retrieval Workshop* , p. 37–46. December 6, Belgium – Leuven.

1. Vuurpijl, L. G., Schomaker, L. R. B., and Broek, E. L. van den (2002). Vind(x): Using the user through cooperative annotation. In *Proceedings of the IEEE Eighth International Workshop on Frontiers in Handwriting Recognition,* p. 221–226. August 6–8, Canada - Niagara-on-the-Lake.

Various

4. Broek, E. L. van den (2005). *Human-Centered Content-Based Image Retrieval* . PhD-thesis Faculty of Social Sciences, Radboud University Nijmegen, The Netherlands – Nijmegen.
3. Broek, E. L. van den (2002). Neurale Netwerken. In B. T. M. Hofstede (Ed.) *Practicum handleiding psychologische functieleer* , p. 46–51. Nijmegen, Nederland: Radboud Universiteit Nijmegen - Universitair Onderwijsinstituut Psychologie en Kunstmatige Intelligentie.
2. Broek, E. L. van den (2001). *Diagnosis of emotional state using pitch analysis and the SUD: An exploratory feasibility study* . Master of Science thesis Faculty of Social Sciences, Radboud University Nijmegen, The Netherlands – Nijmegen.
1. Broek, E. L. van den, Blijleven, M. I., Coenen, A. M. L., Haselager, W. F. G., Kamsteeg, P. A., and Vugt, H. C. van (2001). *Zelfstudie Cognitiewetenschap (Kunstmatige Intelligentie) 2001* . Nijmegen, Nederland: Drukkerij MacDonald.

Abstracts

10. Broek, E. L. van den, Jonker, C. M., Sharpanskykh, A., Treur, J., and Yolum, p. (2005). Formal modeling and analysis of organizations.* *[submitted]*

9. Broek, E. L. van den, Rikxoort, E. M. van, Kisters, P. M. F., Schouten, Th. E., and Vuurpijl, L. G. (2005). Human-Centered Computer Vision. Poster presentation at the *Symposium on the Art and Science of Pattern Recognition* . April 18, The Netherlands – Delft.

8. Broek, E. L. van den, Rikxoort, E. M. van, Kisters, P. M. F., Schouten, Th. E., and Vuurpijl, L. G. (2005). Human-Centered Content-Based Image Retrieval [Poster]. In C. Klöditz (Ed.), *Proceedings of fourth NWO ToKeN symposium* , p. 5. March 18, The Netherlands – Eindhoven.

7. Israël, M., Broek, E. L. van den, Putten, P. van der, and Uyl, M. J. den (2004). Real time automatic scene classification. In R. Verbrugge and L. R. B. Schomaker and N. Taatgen (Eds.), *Proceedings of the Belgian Dutch Artificial Intelligence Conference (BNAIC) 2004* , p. 401–402. October 21–22, Groningen – The Netherlands.

6. Broek, E. L. van den and Bergboer, N. H. (2004). User-centered intelligent content-based image retrieval. In C. Klöditz (Ed.), *Proceedings of third NWO ToKeN2000 symposium* , p. 5. March 26, The Netherlands – Groningen.

5. Broek, E. L. van den (2004). Eidetic: Intelligent Content-Based Image Retrieval. *NICI Annual report 2003,* The Netherlands – Nijmegen.

4. Broek, E. L. van den, Vuurpijl, L. G., Hendriks, M. A., and Kok, T. (2003). Human-Centered Content-Based Image Retrieval. In G. Strube and R. Malaka (Eds.), *Program of the Interdisciplinary College 2003: Applications, Brains & Computers* , p. 39–40. March 7–14, Germany – Günne, Möhnesee.

3. Broek, E. L. van den and Vuurpijl, L. G. (2003). Human-Centered Content-Based Image Retrieval: The emphasis on color. In C. Klöditz (Ed.), *Proceedings of the second NWO ToKeN2000 symposium* , p. 22. February 21, The Netherlands – Delft.

2. Broek, E. L. van den (2003). The pitch of the voice: An indirect physiological measure. In A. Bolt, I. van der Craats, M. van den Noort, A. Orgassa, and J. Stoep (Eds.), *Program of the Association Néerlandaise de Linguistique Appliquée junior day* ; Vol. 13 (pp. A1). The Netherlands – Nijmegen.

1. Broek, E. L. van den (2001). Pitch analysis: An indirect physiological stress measure. In P.A. Starreveld (Ed.), *Proceedings of the Biennial winter conference of the Dutch Society for Psychonomics;* Vol.8 (pp. 60–61). Leiden: NVP.

Curriculum Vitae

Egon L. van den Broek werd geboren op 22 augustus 1974 te Nijmegen. In 2001 studeerde hij af in artificiële intelligentie (AI) aan de Radboud Universiteit Nijmegen onder supervisie van Dr. Ton Dijksta. Zijn afstuderen betrof de ontwikkeling van een methode om emoties te meten door spraakanalyse. Van eind 2001 tot eind 2004 deed hij als NWO promovendus onderzoek aan het Nijmegen Instituut voor Cognitie en Informatie (NICI) onder supervisie van Prof. dr. Charles de Weert, Dr. Louis Vuurpijl en Dr. Theo Schouten. Binnen dit onderzoek zijn op de mens geïnspireerde technieken ontwikkeld voor een nieuwe generatie zoekmachines die op basis van kleur, patronen en vormen in beeldmateriaal zoeken naar vergelijkbaar beeldmateriaal. Dit heeft onder meer geleid tot een unieke kleurruimtesegmentatie, diverse beeldverwerkingstechnieken, een online benchmark voor Content-Based Image Retrieval systemen en een online Multimedia for Art ReTrieval (M4ART) systeem. Hij is auteur van meer dan 30 wetenschappelijke publicaties en heeft meer dan 10 studenten bij hun afstudeerproject begeleid. Zijn onderzoeksinteresses betreffen onder andere cognitiewetenschap, beeldverwerking, agent technologie, mens-machine interactie en validatie en methoden van onderzoek. Sinds oktober 2004 is hij aangesteld als universitair docent AI aan de Vrije Universiteit Amsterdam.